VERITAS Media Manager™
Device Configuration Guide

Release 3.2

UNIX
May 1999
P/N 100-001009

© 1995-1999 VERITAS® Software Corporation. All rights reserved.

Portions of this software are derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm. Copyright 1991-92, RSA Data Security, Inc. Created 1991. All rights reserved.

TRADEMARKS

VERITAS, VxVM, VxVA, VxFS, FirstWatch, and the VERITAS logo are registered trademarks of VERITAS Software Corporation in the United States and other countries.

VERITAS Volume Manager, VERITAS File System, VERITAS NetBackup, VERITAS HSM, VERITAS Media Librarian, CVM, VERITAS Quick I/O, and VxSmartSync are trademarks of VERITAS Software Corporation.

Other products mentioned in this document are trademarks or registered trademarks of their respective holders.

Contents

About This Guide . . . . . **xi**

1. Auspex **1**

Configuring Robotic Controls 2

Configuring SCSI Robotic Controls 2

The SCSA Generic driver 2

Loading the SCSA Generic driver 3

Examples of SCSI Robotic Control Device Files 5

Configuring RS-232 Robotic Controls 6

Examples of RS-232 Robotic Control Device Files. 7

Changing SCSI ID Mapping in Kernel Configuration 8

Finding the SunOS Kernel Configuration File 8

Checking the SCSI Device Unit Assignment Table 9

Changing the SCSI Device Unit Assignment Table 9

Logical Unit Numbers 9

Configuring Tape Drives 11

Creating Device Files 11

Examples of No Rewind Device Files 12

Adding Nonstandard Tape Drives 13

Note on Case and Spaces in `st_conf.c` Entries 13

Adding Exabyte Compression Drives 13

Adding HP 4-mm Drives and HP C1560B DAT Autoloaders . . 15

Adding STK Drives 17

Adding Quantum DLT Drives or Stackers 18

Configuring HP Optical Disk Drives 20

Creating Device Files 20

Setting the Optical Drive Type in Nonvolatile Memory 21

Rebuilding a SunOS Kernel 22

Command Summary 24

2. Sun4/SPARC Running Solaris 2.5/2.5.1/2.6/7.0 27

Installing SCSI Pass-Through Drivers 29

Configuring the SG Driver 30

Configuring Robotic Controls 33

Configuring SCSI Robotic Controls 33

Examples of SCSI Robotic Control Device Files 34

Configuring RS-232 Robotic Controls 36

Examples of RS-232 Robotic Control Device Files 36

Configuring Tape Drives 37

Using Berkeley-Style Close 37

Fast-Tape Positioning (locate-block) 37

Enabling locate-block 37

Disabling locate-block 37

No Rewind Device Files 38

Examples of No Rewind Device Files 38

Configuring Nonstandard Tape Drives 39

Note on Case and Spaces in `st.conf` Entries 40

NetBackup 3.2 Media Manager Device Configuration Guide for UNIX

Additions to `st.conf` 40
Adding Logical Unit Number Entries 41
Multiple Logical Unit Numbers per SCSI Target. 42
Adding HP 4-mm Drives and HP DAT Autoloaders 45
Adding Sony AIT Drives 45
Configuring HP Optical Disk Drives 48
Creating Device Files 48
Examples of Optical Disk Device Files 49
Setting the HP Optical Drive Type in Nonvolatile Memory 49
Command Summary. 51
3. IBM RS6000 Running AIX 4.1.5/4.2/4.3/4.3.1/4.3.2 53
RS6000 AIX Adapter Number Conventions 55
Installing the SCSI Pass-Through Driver 56
Configuring Robotic Controls 57
Configuring SCSI Robotic Controls 57
Examples of SCSI Robotic Control Device Files 59
Configuring IBM 3590 Stacker Robotic Controls 62
Configuring RS-232 Robotic Controls 62
Example of RS-232 Robotic Control Device Files 63
Configuring Tape Drives 65
Configuring Non-QIC Tape Drives 65
Using Extended-File Marks for Drives 65
Fast-Tape Positioning (locate-block) 66
Creating Device Files 66
No Rewind Device File Example 68
Using Multiple Tape Densities 69
Adding HP 4-mm Drives and HP C1560B DAT Autoloaders. 70

Adding Sony AIT Drives 70
No Rewind Device Files 70
Dip Switch Settings 70
Configuring Optical Disk Drives 73
Creating Device Files 73
Examples of Optical Disk Device Files 74
Setting an HP Optical Drive Type in Nonvolatile Memory 75
Command Summary 79
4. HP9000-700 Running HP-UX 10.20/11.0 81
Configuring Robotic Controls 82
Configuring SCSI Robotic Controls 82
Examples of SCSI Robotic Control Device Files 83
Configuring RS-232 Robotic Controls 84
Example of RS-232 Robotic Control Device File 85
Configuring Tape Drives 86
Using Berkeley Style Close 86
Device Files 86
Examples of No Rewind Device Files 87
Switch Settings for HP C1533A 4-mm DAT Drives 89
Configuring Optical Disk Drives 90
Examples of Optical Disk Device Files 90
Command Summary 92
5. HP9000-800 Running HP-UX 10.20/11.0 93
Configuring Robotic Controls 94
Configuring SCSI Robotic Controls 94
Determining Which Pass-Through Driver to Configure 94
Configuring Device Files for spt Pass-Through Driver 95

Configure Device Files for sctl Pass-Through Driver 97
Configuring RS-232 Robotic Controls 100
Example of RS-232 Robotic Control Device File 100
Configuring Tape Drives 101
Using Berkeley Style Close 101
Device Files 101
Example of No Rewind Device File 102
Fast-Tape Positioning (locate-block) 102
Switch Settings for HP C1533A 4-mm DAT Drives 104
Configuring Optical Disk Drives 105
Example of an Optical Disk Device File 105
Command Summary 106
6. IRIX 6.2 107
Using SCIP Controllers 109
Note on the `mediad` Command 110
Configuring Robotic Controls 111
Configuring SCSI Robotic Controls 111
Examples of SCSI Robotic Control Device Files 112
Configuring RS-232 Robotic Controls 113
Examples of RS-232 Robotic Control Device Files 113
Configuring Tape Drives 114
Fast-Tape Positioning (locate-block) 114
Creating Device Files 114
Examples of No Rewind Device Files 115
Adding Exabyte Compression Drives 116
Changing `/var/sysgen/master.d/scsi` 116
Reconfiguring the Kernel and Modifying `MAKEDEV` 117

Adding HP 4-mm Drives and HP C1560B DAT Autoloaders. 117
Checking Switch Settings 117
Changing /var/sysgen/master.d/scsi 118
Adding Quantum DLT Drives or Stackers 119
Adding Sony DTF Drives 121
Configuring Optical Disk Drives 122
Example of Optical Disk Device Files. 122
Command Summary. 123
7. IRIX 6.4/6.5 125
Using SCIP Controllers. 127
Note on the mediad Command 128
Configuring Robotic Controls 129
Configuring SCSI Robotic Controls 129
Examples of SCSI Robot Control Device Files 129
Configuring RS-232 Robotic Controls 130
Examples of RS-232 Robot Control Device Files 130
Configuring Tape Drives 132
Examples of No Rewind Device Files 132
Adding HP 4-mm Drives and HP C1560B DAT Autoloaders. 133
Checking Switch Settings 133
Changing /var/sysgen/master.d/scsi 134
Fast-Tape Positioning (locate-block) 135
Adding Sony DTF Drives 136
Configuring Optical Disk Drives 137
Examples of Optical Disk Device Files 137
Command Summary. 138
8. DEC Alpha Running Digital UNIX 4.0878 139

NetBackup 3.2 Media Manager Device Configuration Guide for UNIX

Configuring Robotic Controls 140
Configuring SCSI Robotic Controls 140
Examples of SCSI Robotic Control Device Files 141
Configuring RS-232 Robotic Controls 142
Example of RS-232 Robotic Control Device Files 142
Configuring Tape Drives 143
Fast-Tape Positioning (locate-block) 143
Adding Standard Tape Drives 143
Examples of No Rewind Device Files 144
Adding Nonstandard Tape Drives 145
Switch Settings for HP C1533A 4mm DAT Drives 145
Command Summary 147
9. NCR Running MP-RAS 3.02 149
NCR Device Files 150
Configuring Robotic Controls 151
Configuring Tape Drives 151
10. Sequent Running DYNIX/ptx 4.2.1/4.2.3/4.4.2/4.4.4 153
Configuring Robotic Controls 154
Configuring SCSI Robotic Controls 154
Configuring Tape Drives 156
Kernel Configuration 157
Turning Off Messages 157
Exabyte Drive Type 157
DLT Drive Type 157
Tape Drive Support 158
DLT Drive Type 158
IBM Magstar (3590) Drive Type 158

Command Summary 159

11. Pyramid RM1000 Running Reliant UNIX 5.43 B0078 161

Configuring Robotic Controls 162

Configuring Tape Drives 162

Using Berkeley-style Close 162

Checking For Berkeley-style Close 162

Setting Berkeley-style Close 163

Glossary 165

Index 181

NetBackup 3.2 Media Manager Device Configuration Guide for UNIX

About This Guide

Introduction

This guide contains information on topics that VERITAS has found useful when adding storage peripherals controlled by Media Manager to Media Manager hosts. Media Manager is the component of NetBackup and Storage Migrator that manages devices and media.

Audience

The intended audience is the system administrator responsible for adding the storage peripherals and assumes a thorough knowledge of UNIX system and device configuration.

Scope

The information in this guide supplements the manuals provided by hardware and operating system vendors.

Organization

There is a separate chapter for each of the supported server platforms. The chapters are organized as follows

- Chapter 1, “Auspex”
- Chapter 2, “Sun4/SPARC Running Solaris 2.5/2.5.1/2.6/7.0”

- Chapter 3, “IBM RS6000 Running AIX 4.1.5/4.2/4.3/4.3.1/4.3.2”
- Chapter 4, “HP9000-700 Running HP-UX 10.20/11.0”
- Chapter 5, “HP9000-800 Running HP-UX 10.20/11.0”
- Chapter 6, “IRIX 6.2”
- Chapter 7, “IRIX 6.4/6.5”
- Chapter 8, “DEC Alpha Running Digital UNIX 4.0878”
- Chapter 9, “NCR Running MP-RAS 3.02”
- Chapter 10, “Sequent Running DYNIX/ptx 4.2.1/4.2.3/4.4.2/4.4.4”
- Chapter 11, “Pyramid RM1000 Running Reliant UNIX 5.43 B0078”

In addition to these chapters, there is a glossary for terms that you may encounter when using and discussing NetBackup; and an index.

Using This Guide

Each UNIX platform supported by VERITAS Media Manager is described in a separate chapter in this guide. You should have to use only the chapters for the platforms on which you are configuring devices to be under Media Manager control.

Related Documents

Other documents that will be useful when configuring the VERITAS Media Manager are listed below.

- *NetBackup Release Notes - UNIX*

  Contains important information on new features and supported platforms and operating systems. The release notes also contain information on operation, known problems, and problems fixed that may not be in the manuals.

- *NetBackup Release Notes - Windows NT Server*

  Contains important information on new features and supported platforms and operating systems. The release notes also contain information on operation, known problems, and problems fixed that may not be in the manuals.

- *NetBackup Troubleshooting Guide - UNIX*

  Provides troubleshooting information for NetBackup.

- *NetBackup Installation Guide - UNIX*

  Provides instructions about modifying scripts.

- *Storage Migrator Release Notes - UNIX*

  Provides information such as, the platforms and operating systems that are supported and operating notes that may not be in the Storage Manager manuals.

- *Storage Migrator System Administrator's Guide - UNIX*

  Explains how to configure and manage Storage Manager on a UNIX system.

Depending on your configuration, other documents may also be required. For a complete list of documents pertaining to NetBackup, see either of the NetBackup release notes.

Conventions

The following table describes the typographic conventions used in this guide.

Typeface	Usage	Examples

`courier`	Computer output; user input; names of commands, files, and directories	`$ You have mail.`
		The `mkdir` command displays files.
		`$ ls -a`

italics	New terms; document titles; words to be emphasized; variables to be substituted with a real name or value	`$ mkdir` *filename* Refer to the *User's Guide* for details.

Getting Help

For license information or information about VERITAS service packages, contact VERITAS Customer Support.

US Customers: 1-800-342-0652

International Customers: +1 (650) 335-8555

Fax: (650) 335-8428

VERITAS Customer Support can also be reached through electronic mail at:

`support@veritas.com`

Auspex

This chapter provides information for configuring devices on an Auspex server running SunOS 4.1.3/4.1.4. You configure drives and robots using one of the available Media Manager administrative interfaces.

Typical device path names you enter when configuring drives and robots are described in this chapter. Instructions for changing and rebuilding the kernel are also given. Depending on the type and number of devices you are adding, you may have to enter information into kernel source files and then reconfigure the kernel.

You can copy code examples from the on-line version of this manual, located in `/usr/openv/volmgr/dev_config_guide.txt`. This file is installed with Media Manager software.

This main topics in this chapter are as follows:

- Configuring Robotic Controls
- Changing SCSI ID Mapping in Kernel Configuration
- Configuring Tape Drives
- Configuring HP Optical Disk Drives
- Rebuilding a SunOS Kernel
- Command Summary

Note: VERITAS does not recommend or support the use of single-ended to differential SCSI converters on Media Manager controlled devices. You may encounter problems if you use these converters.

Configuring Robotic Controls

Robots can be controlled through a SCSI, an RS-232, or a network connection. Configuration for network controlled robotic libraries is discussed in the appendices of the *Media Manager System Administrator's Guide - UNIX*. SCSI and RS-232 control is covered in the following sections.

Configuring SCSI Robotic Controls

Read this topic if you plan to use a robotic storage device that is controlled through a SCSI robotic connection.

Supported SCSI robots include the following. See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with these robot types.

- ODL - Optical Disk Library
- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TL4 - Tape Library 4MM
- TL8 - Tape Library 8MM
- TLD - Tape Library DLT
- TS8 - Tape Stacker 8MM
- TSD - Tape Stacker DLT

SCSI robotics are supported on Auspex systems with sun4c or sun4m kernel architecture. SCSI robotics are not supported on systems with sun4 kernel architecture. To determine the kernel architecture, you can use the `/usr/bin/arch -k` command.

The SCSA Generic driver

The SCSA Generic (SG) driver is a loadable driver used in combination with Media Manager robotic software to control SCSI robotic peripherals. When installing SCSI-controlled robotic software on a server running SunOS, you need to install this driver to use the peripheral's robotic control.

If the only robotics you have are on an Auspex Storage Processor (SP), you do not need to load the SG driver. The passthru driver for robotics on a SP is built into the system.

Since the SG driver is loadable, the kernel does not have to be reconfigured and the system does not have to be rebooted to install this driver. However, the driver must be installed and reloaded each time the system is booted and VERITAS recommends that you automate this procedure (for example, by putting it in `/etc/rc.local`).

Loading the SCSA Generic driver

The following instructions explain how to load the SG driver. You must perform these steps as the root user.

1. Determine what loadable kernel modules are currently loaded by executing the `modstat` command:

   ```
   /usr/etc/modstat
   Id  Type  Loadaddr  Size   B-major   C-major   Sysnum Mod Name
   1   Drv   ff08f000  5000             59.       SCSA Generic Driver
   <no output is produced if no loadable drivers are present>
   ```

   - If an SG driver is already installed (as in the above example), you must unload it before trying to install the new SG driver. Refer to step 2.
   - If the `modstat` output shows any other loadable drivers, ensure that they are not used for communicating with the same SCSI robotic devices that Media Manager will access through the SG driver. If there are any such drivers, remove them as explained in step 2. A case where a conflicting driver could exist is where it is from another backup product.
   - If there is no SG or other conflicting driver installed, proceed to step 3.

2. Unload an existing SG or other loadable driver using the `modunload` command.

   The following is an example of how to unload the SG driver. The `-id` value that you use with `modunload` is the Id number of the driver as shown by `modstat`.

   ```
   /usr/etc/modstat
   ```

   ```
   Id  Type  Loadaddr   Size   B-major   C-major  Sysnum Mod Name
   1   Drv   ff08f000   5000             59.      SCSA Generic Driver
   ```

   ```
   /usr/etc/modunload -id 1
   /usr/etc/modstat
   ```

   ```
   <no output is produced if no loadable drivers are present>
   ```

3. Run the SG driver installation script provided with Media Manager by entering the following:

   ```
   /usr/openv/volmgr/bin/driver/sg.install
   ```

   This script loads the appropriate SG driver based on the system's kernel architecture and creates the `/dev/sg` device files.

4. Verify that the driver was loaded, using the `modstat` command.

   ```
   /usr/etc/modstat
   ```

   The output should be similar to the following:

   ```
   Id  Type  Loadaddr   Size   B-major   C-major  Sysnum Mod Name
   1   Drv   ff08f000   5000             59.      SCSA Generic Driver
   ```

5. The driver must be installed each time the system is booted. To install the SG driver at boot time on systems running SunOS, the following code can be placed in the `/etc/rc.local` start up script.

   ```
   Install the SG driver

   if [ -f /usr/openv/volmgr/bin/driver/sg.install ]
   then
        (cd /usr/openv/volmgr/bin/driver; ./sg.install)
   else
        echo "sg driver not installed." > /dev/console
   fi
   ```

Note: To display SCSI inquiry strings for devices available through the SG driver, execute `/usr/openv/volmgr/bin/sgscan`.
On Auspex, to display SCSI inquiry strings for `/dev/asc*` devices, execute `/usr/openv/volmgr/bin/spscan`.

Examples of SCSI Robotic Control Device Files

Example 1:

On SunOS systems, SCSI controlled robotics use device files located in the `/dev/sg` directory. The format of the device file paths follows:

/dev/sg/c*Controller*t*Target*d*Lun*s0

Where:

Controller is the SCSI bus (adapter) number

Target is the SCSI ID

Lun is the logical unit number and is always 0 (except for some peripherals, such as DLT2700, DLT4700, and HP C1560B).

For example:

- If the robotics control is not for a DLT2700, DLT4700, HP C1560B, or other LUN 1 peripheral and is on SCSI bus (adapter) 0 at SCSI ID 5, the device file you specify is:

  `/dev/sg/c0t5d0s0`

- If the robotics control is not for a DLT2700, DLT4700, HP C1560B, or other LUN 1 peripheral and is on SCSI bus (adapter) 1 at SCSI ID 3, the device file you specify is:

  `/dev/sg/c1t3d0s0`

- If a DLT2700, DLT4700, HP C1560B, or other LUN 1 peripheral robotics control is on SCSI bus (adapter) 0 at SCSI ID 4 with logical unit number 1, the device file you specify is:

  `/dev/sg/c0t4d1s0`

Example 2:

If the robotic device is connected to an Auspex SP, the format of the device file path follows:

`/dev/asc`*xx*

Where *xx* is the slot number within the SP. Slot numbers can be determined by running the `/usr/openv/volmgr/bin/spscan` script.

For example.

An Odetics ATL 4/52 with the robotics connected to slot 38 would have a the following robotic path:

```
/dev/asc38
```

Example 3:

If a Quantum DLT4700 is being used on an Auspex SP, a special case file must be created indicating to the TSD software that LUN 1 must be used when communicating with the robotics. If the slot number of the DLT4700 is 40, the device file for robotics is `/dev/asc40`. The following command must also be used:

```
touch /dev/asc40.1
```

Configuring RS-232 Robotic Controls

Note: RS-232 robotic controls are not supported on the Auspex Storage Processor.

Read this topic if you plan to use a robotic storage device that is controlled through an RS-232 connection. Supported RS-232 robots include:

- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TLD - Tape Library DLT

See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with the above robot types.

Configuring RS-232 controlled devices on a SunOS platform normally consists of choosing between the A and B serial console ports.

Examples of RS-232 Robotic Control Device Files

The following are examples of the path names you use when adding an RS-232 controlled robotic device to a Media Manager configuration:

```
serial port A:  /dev/ttya
serial port B:  /dev/ttyb
```

Changing SCSI ID Mapping in Kernel Configuration

Read this topic if you have not yet verified that the kernel configuration file for SunOS on this system supports the number of tape and optical drives you have connected and the SCSI IDs for those devices.

When installing Media Manager and robotic software, you may need to reconfigure the SunOS kernel to support the number of tape or optical drives being added or to support a different SCSI ID. The data path to SCSI tape drives goes through the `st(4s)` SCSI tape driver, while the optical drives are used through the `sd(4s)` SCSI disk driver.

Finding the SunOS Kernel Configuration File

The kernel configuration file contains a table of SCSI device unit assignments. This file is located in:

`/usr/sys/`*arch*`/conf/`*file*

Where:

arch is the kernel architecture for the system and can be determined using the `arch -k` command.

file is the configuration file for the running system.

The configuration file for the running SunOS can normally be determined by examining the `/etc/motd` file. For example, the following `/etc/motd` file shows that the kernel name is GENERIC.

```
cat /etc/motd
```

```
SunOS Release 4.1.3 (GENERIC) #7:Mon Dec 13 09:58:55 CST 1993
```

An alternate method for determining the kernel name is as follows:

```
strings /vmunix | grep SunOS
```

```
SunOS Release 4.1.3 (GENERIC) #7:Mon Dec 13 09:58:55 CST 1993
```

Using the above example, the kernel configuration file could be:

```
/usr/sys/sun4m/conf/GENERIC
```

Checking the SCSI Device Unit Assignment Table

Within the SunOS kernel configuration file is a table of SCSI device unit assignments that maps the SCSI bus, target, and logical unit number of a device to a tape or disk number for the corresponding device driver (st or sd). This table is located near the end of the kernel configuration file.

The following is a portion of a sample SCSI device unit assignment table:

```
scsibus0 at esp # declare first SCSI bus
scsibus1 at esp # declare second SCSI bus
disk sd3 at scsibus0 target 0 lun 0   # first SCSI disk
disk sd1 at scsibus0 target 1 lun 0   # second SCSI disk
tape st0 at scsibus0 target 4 lun 0   # first SCSI tape
tape st1 at scsibus0 target 5 lun 0   # second SCSI tape
tape st2 at scsibus1 target 4 lun 0   # third SCSI tape
tape st3 at scsibus1 target 5 lun 0   # fourth SCSI tape
```

Changing the SCSI Device Unit Assignment Table

In the above example, the first SCSI tape device, st0, is declared to be attached to the first SCSI bus, at SCSI ID (target) 4, and logical unit number (lun) 0. The disk device sd3 is declared to be attached to the first SCSI bus, at SCSI ID (target) 0, and logical unit number (lun) 0.

You may have to change this table, depending on the SCSI bus and SCSI ID of the tape or optical drive. If you change this table, the kernel has to be reconfigured and rebuilt to recognize the changes. See “Rebuilding a SunOS Kernel” on page 22 for an example of how to reconfigure and rebuild a SunOS kernel. Before rebuilding the kernel, you should read the other topics to see if additional changes are necessary because of the type of the tape or optical drive.

Logical Unit Numbers

Tape devices (such as HP C1560B DAT Autoloaders or STK half-inch cartridge drives) that use the logical unit number characteristic require special attention. When devices use a logical unit number, multiple drives all share the same SCSI ID (target) and are differentiated only by their logical unit number at that specific SCSI target.

The following is a portion of a sample SCSI device unit assignment table that employs logical unit numbers:

```
scsibus0 at esp # declare first SCSI bus
scsibus1 at esp # declare second SCSI bus
disk sd3 at scsibus0 target 0 lun 0   # first SCSI disk
disk sd1 at scsibus0 target 1 lun 0   # second SCSI disk
tape st1 at scsibus1 target 3 lun 0   # first SCSI tape
tape st2 at scsibus1 target 3 lun 1   # second SCSI tape
tape st3 at scsibus1 target 3 lun 2   # third SCSI tape
tape st4 at scsibus1 target 3 lun 3   # fourth SCSI tape
```

In this example:

- The first SCSI tape device, `st1`, is declared to be attached to the SCSI bus 1, at SCSI ID (target) 3, and logical unit number 0.
- The second (`st2`), third (`st3`), and fourth (`st4`) tape drives are also attached to SCSI bus 1 at SCSI ID (target) 3.

The distinguishing characteristic of these four drives is their logical unit number.

Note: The HP C1560B DAT Autoloader always uses a logical unit number of 1.

Configuring Tape Drives

When adding tape drives to a Media Manager configuration, you must specify a no rewind on close device path. In a typical SunOS configuration, most of the desired tape device files already exist in the `/dev` directory. These device files have the following format:

`/dev/nrst`*ST_Number*+*Density*

Where:

ST_Number is the tape device number configured to the desired SCSI bus and SCSI ID in the kernel configuration file.

Density is 0, 8, or 16, depending on the drive's density capabilities. *Density* is added to *ST_Number.*

For Exabyte drives, a density of

- 0 is added to the device number for 8200 drives
- 8 is added to the device number for 8500 drives
- 16 is added to the device number for 8500C and 8505 drives

Other drive types normally use 0 for the density, unless multiple densities are specified in the `st_conf.c` file. (Refer to the `st(4S)` man page.)

Creating Device Files

If the required device files do not exist, you can use the `MAKEDEV` command to create device files for a particular SCSI tape number as follows:

```
cd /dev
MAKEDEV st ST_Number
```

Where *ST_Number* is the tape device number assigned to the desired SCSI bus and SCSI ID in the SCSI device unit assignment table (see "Checking the SCSI Device Unit Assignment Table" on page 9).

For example, if the desired tape drive is on SCSI bus 1 at SCSI ID 3 and the SCSI device unit assignment table contains the following line:

```
tape st7 at scsibus1 target 3 lun 0 # tape drive
```

the tape device number is 7. The following commands create the device file:

```
cd /dev
MAKEDEV st7
```

If the tape drive is connected to an Auspex SP, the no rewind on close device file for the drive follows:

`/dev/nrast`*slot_number*

Where *slot_number* is the slot number.

Examples of No Rewind Device Files

Example 1:

If the desired Exabyte tape drive is on SCSI bus 1 at SCSI ID 3 and the SCSI device unit assignment table contains the following line:

```
tape st7 at scsibus1 target 3 lun 0     # tape drive
```

then the *ST_Number* is 7 and the path would be one of following (depending on the type of Exabyte drive):

`/dev/nrst7` (Exabyte 8200)

`/dev/nrst15` (Exabyte 8500)

`/dev/nrst23` (Exabyte 8500C or 8505)

Example 2:

If the desired 4-mm (DAT) tape drive is on SCSI bus 0 at SCSI ID 3, and the kernel configuration file contains the following line:

```
tape st1 at scsibus0 target 5 lun 0 # tape drive
```

then the *ST_Number* is 1, and the device path follows:

```
/dev/nrst1
```

Example 3:

On an Auspex SP, a DLT tape drive is connected to slot 39, as determined using `/usr/openv/volmgr/bin/spscan`. For example, if this command returns the following output:

```
/dev/asc39: removable dev type 1h Quantum DLT4000 CC1E
```

then the device path is as follows:

```
/dev/nrast39
```

Adding Nonstandard Tape Drives

Adding any of the drives mentioned in this section may require you to modify and rebuild the SunOS kernel. The following topics explain how to determine if kernel changes are necessary and how to make those changes.

Note on Case and Spaces in `st_conf.c` Entries

Upper and lower case are significant. For example, using QUANTUM instead of Quantum would not work for DLT4000 drives.

Spaces are significant within quoted strings in the `st_conf.c` file. For example, if the first part of the entry for an HP C1533A drive is as follows (string length of 14, including spaces):

```
14, "HP      C1533A"
```

And you were to omit some of the spaces, as in the following (string length of nine, including spaces), the drive would not be recognized correctly.

```
14 "HP C1533A"
```

The best way to ensure that your entries are accurate is to copy them from the on-line version of this manual whenever possible.

Adding Exabyte Compression Drives

Read this topic if you plan to use one or more standalone or robotic Exabyte compression drives (8500C, 8505, 8505XL, 8900). This topic is also important if you want to take advantage of faster file-skip performance on non-compression Exabyte tape drives (see the text on ST_KNOWS_EOD in step 1 below).

You may have to modify and rebuild the SunOS kernel for the system to recognize the Exabyte 8500C, 8505, 8505XL, 8900 drives. The following procedure explains the steps you should perform.

1. Check if the following code exists in the `struct st_drivetype st_drivetypes[]` array in the `/sys/scsi/targets/st_conf.c` file.

```
/* Exabyte 8mm half-height compression cartridge 8505 or 8505XL */
{
    "Exabyte EXB-8505 8mm Helical Scan", 16, "EXABYTE EXB-8505",
     ST_TYPE_EXB8505, 1024,
     (ST_VARIABLE | ST_BSF | ST_BSR | ST_LONG_ERASE |
     ST_KNOWS_EOD ),
     5000, 5000,
     { 0x14, 0x15, 0x00, 0x8C },
     {  0, 0, 0, 0 }
},
/* Exabyte 8mm compression cartridge */
{
    "Exabyte EXB-8500C 8mm Helical Scan", 16, "EXABYTE EXB8500C",
     ST_TYPE_EXB8500C, 1024,
     (ST_VARIABLE | ST_BSF | ST_BSR | ST_LONG_ERASE |
     ST_KNOWS_EOD ),
     5000, 5000,
     { 0x14, 0x15, 0x00, 0x8C },
     {  0, 0, 0, 0 }
},
/* Exabyte 8mm compression cartridge */
{
    "Exabyte EXB-8900 Mammoth", 16, "EXABYTE EXB-8900",
     ST_TYPE_EXB8505, 1024,
     (ST_VARIABLE | ST_BSF | ST_BSR | ST_LONG_ERASE |
     ST_KNOWS_EOD ),
     5000, 5000,
     { 0x27, 0x27, 0x27, 0x00 },
     {  0, 0, 0, 0 }
},
```

   Drives may have different vendor/product strings than the strings shown here. In the example above for an Exabyte 8505, `"EXABYTE EXB-8505"` is the vendor/product string. The 16 preceding this string is the string length and must compare. To view the vendor/product strings for your drives, you can use the `dmesg(8)`command shortly after boot. The vendor and product strings for a drive are also logged with the `syslogd(8)` utility when the system is booted. The `syslogd(8)` utility typically logs to `/var/adm/messages`.

CAUTION! Always save a copy of a kernel file before changing it. This allows you to easily restore the file in case you make a mistake.

If this code is not in the `/sys/scsi/targets/st_conf.c` file, add it. The best way to do this is to copy it from the `dev_config_quide.txt` file.

For better file-skip performance on Exabyte drives, you may also want to add the ST_KNOWS_EOD attribute (as specified in the example code above) to the `st_conf.c` file for all Exabyte drive types. The `st_conf.c` file included in the standard SunOS does not contain this attribute for any Exabyte drive types.

2. Check for the following lines in `/sys/scsi/targets/stdef.h`:

```
define ST_TYPE_EXB8505 0x31 /*Exabyte 8505,8905XL,or
8900*/

define ST_TYPE_EXB8500C 0x32 /*Exabyte 8500C */
```

   If these lines are not in `stdef.h`, add them.

3. If you changed the `st_conf.c` or `stdef.h` files, you will have to rebuild the kernel and then reboot the system for any of these changes to become effective. Do this after completing all other necessary changes to the kernel. See “Rebuilding a SunOS Kernel” on page 22 for instructions.

Adding HP 4-mm Drives and HP C1560B DAT Autoloaders

Read this topic if you plan to use one or more standalone or robotic Hewlett-Packard (HP) 4-mm DAT tape drives or HP C1560B DAT Autoloaders. It explains drive switch settings and SunOS kernel changes you may have to make in order for the system to recognize these drives.

First, ensure that the hardware switch settings on the drives are as follows. Other switch combinations may work, but these are the settings that were functional during testing with an HP 35480A drive and an HP C1560B Autoloader.

On=1, Off=0

Switch	Setting
1	1
2	1
3	1
4	1
5	1

Switch	Setting
6	1
7	0
8	0

You may also have to make changes to the SunOS kernel and then rebuild it. The following explains how to determine if changes are necessary and how to make them.

CAUTION! Always save a copy of a kernel file before changing it. This allows you to easily restore the file.

1. Check if the following code is in the `struct st_drivetype st_drivetypes[]` array in the `/sys/scsi/targets/st_conf.c` file.

```
/* HP 4mm Helical Scan Tape */
{
   "HP 4mm DAT", 13, "HP      HP354", ST_TYPE_HP1, 10240,
    (ST_VARIABLE | ST_BSF | ST_BSR | ST_LONG_ERASE |
    ST_KNOWS_EOD),
    6000, 6000,
    { 0x00, 0x00, 0x00, 0x00},
    { 0, 0, 0, 0 }
},

/* HP C1560B DAT Autoloader */
{
   "HP DAT Autoloader", 13, "HP      C1533", ST_TYPE_HP1, 10240,
    (ST_VARIABLE | ST_BSF | ST_BSR | ST_LONG_ERASE |
    ST_KNOWS_EOD),
    6000, 6000,
    { 0x00, 0x00, 0x00, 0x00},
    { 0, 0, 0, 0 }
},
```

   Drives may have different vendor/product strings than the strings shown here. In the example above for an HP 4-mm, `"HP      HP354"` is the vendor/product string. The 13 preceding this string is the string length and must compare. To view the vendor/product strings for your drives, you can use the `dmesg(8)` command shortly after boot. The vendor and product strings for a drive are also logged with the `syslogd(8)` utility when the system is booted. The `syslogd(8)` utility typically logs to `/var/adm/messages`.

   If this code is not in the `st_conf.c` file, add it. The best way to make this addition is to copy it from `dev_config_quide.txt`.

2. Check for the following lines in `/sys/scsi/targets/stdef.h`:

```
define    ST_TYPE_HP1 0x33 /* HP DAT */
```

If this line is not in `stdef.h`, add it.

3. After completing all other necessary changes to the kernel, rebuild it and reboot the system as explained in “Rebuilding a SunOS Kernel” on page 22. This is necessary for any of these changes to become effective.

Adding STK Drives

Read this topic if you plan to use one or more standalone or robotic StorageTek (STK) half-inch cartridge tape drives. It explains SunOS kernel changes you may have to make in order for the system to recognize these drives.

If the drives are contained in an STK silo, you may need to use multiple logical unit numbers (lun) for a given SCSI ID (target). See “Logical Unit Numbers” on page 9 for a discussion on how to use logical unit numbers.

CAUTION! Always save a copy of a kernel file before changing it. This allows you to easily restore the file if needed.

---

1. Check if the following code is in the `struct st_drivetype st_drivetypes[]` array found in the `/sys/scsi/targets/st_conf.c file`.

```
/* STK 38000 1/2 in cartridge */
{
    "STK", 3, "STK", ST_TYPE_STK, 1024,
     (ST_VARIABLE | ST_BSF | ST_BSR | ST_LONG_ERASE |
      ST_AUTODEN_OVERRIDE | ST_KNOWS_EOD),
      5000, 5000,
      { 0x00, 0x00, 0x00, 0x00 },
      { 0, 0, 0, 0 }
},
```

If this code is not in the `st_conf.c` file, add it. The best way to make this addition is to copy it from `dev_config_quide.txt`.

2. Check for the following line in `/sys/scsi/targets/stdef.h`:

```
define ST_TYPE_STK 0x34 /* STK 1/2 in.Cartridge */
```

If this line is not in `stdef.h`, add it.

3. After completing all other necessary changes to the kernel, rebuild it and reboot the system as explained in “Rebuilding a SunOS Kernel” on page 22. This is necessary for these changes to become effective.

Adding Quantum DLT Drives or Stackers

Read this topic if you plan to use standalone or robotic Quantum DLT2000 or DLT4000 drives or a Quantum DLT2700 or DLT4700 stacker. It explains SunOS kernel changes you may have to make in order for the system to recognize these drives.

CAUTION! Always save a copy of a kernel file before changing it. This allows you to easily restore the file in case you make a mistake.

---

1. Check that the following code is in the `struct st_drivetype st_drivetypes[]` array found in the `/sys/scsi/targets/st_conf.c` file.

```
/* QUANTUM DLT */
{
   "QUANTUM DLT Tape Drive", 15, "Quantum DLT2000",
    ST_TYPE_DLT, 1024,
    (ST_VARIABLE | ST_BSF | ST_BSR | ST_LONG_ERASE |
    ST_AUTODEN_OVERRIDE | ST_KNOWS_EOD),
    5000, 5000,
    { 0x00, 0x00, 0x00, 0x00 },
    {  0, 0, 0, 0 }
},
```

Note: For a DLT4000 drive, create the entry as shown above, except substitute DLT4000 for DLT2000.

For a QUANTUM DLT2700 stacker, add the following to the `struct st_drivetype st_drivetypes[]` array:

```
/* QUANTUM DLT2700 Stacker */
{
   "QUANTUM DLT Tape Drive", 15, "Quantum DLT2700",
    ST_TYPE_DLT, 1024,
    (ST_VARIABLE | ST_BSF | ST_BSR | ST_LONG_ERASE |
    ST_AUTODEN_OVERRIDE | ST_KNOWS_EOD),
    5000, 5000,
    { 0x00, 0x00, 0x00, 0x00 },
    {  0, 0, 0, 0 }
},
```

Note: For a DLT4700 stacker, create the entry as shown above, except substitute DLT4700 for DLT2700.

Devices may have different vendor/product strings than those shown here. In the Quantum DLT2700 drive example, `Quantum DLT2700` is the vendor/product string. The preceding 15 is the string length. It must compare. To view the vendor/product strings for your drives, use the `dmesg(8)` command shortly after boot. The vendor and product strings for a drive are also logged with the `syslogd(8)` utility when the system is booted. The `syslogd(8)` utility typically logs to `/var/adm/messages`.

Note: Some older DLT2000 drives may have DEC instead of Quantum for a vendor ID.

If this code is not in the `st_conf.c` file, add it. The best way to make additions is to copy it from `dev_config_quide.txt`.

2. Check for the following line in `/sys/scsi/targets/stdef.h`:

   ```
   define              ST_TYPE_DLT 0x35 /* Quantum DLT*/
   ```

   If this line is not in `stdef.h`, add it.

3. After completing all other necessary changes to the kernel, rebuild it and reboot the system as explained in “Rebuilding a SunOS Kernel” on page 22. This is necessary for these changes to become effective.

Configuring HP Optical Disk Drives

Note: HP optical disk drives are accessed through the SCSI disk driver (sd). Read the "Setting the Optical Drive Type in Nonvolatile Memory" on page 21 for instructions on configuring the system so this access can occur.

When adding optical disk drives to a Media Manager configuration, you must specify the following device paths:

- Character device path (partition g)
- Volume header device path (partition a)

In a typical SunOS configuration, most of the desired disk device files already exist in the `/dev` directory.

Character disk device files have the following format:

`/dev/rsd`*`sd_number`*`g`

Volume header device files have the following format:

`/dev/rsd`*`sd_number`*`a`

Where:

sd_number is the disk device number configured to the desired SCSI bus and SCSI ID in the kernel configuration file

`g` is the desired disk partition

`a` is the desired disk partition

See the `sd(4S)` man page for further details.

Creating Device Files

If the required device files does not exist, you can use the `MAKEDEV` command to create device files for a particular SCSI optical disk number as follows:

```
cd /dev
MAKEDEV sd sd_number
```

Where *sd_number* is the disk device number assigned to the desired SCSI bus and SCSI ID in the SCSI device unit assignment table (see “Checking the SCSI Device Unit Assignment Table” on page 9).

For example, if the desired optical disk drive is on SCSI bus 1 at SCSI ID 3 and the SCSI device unit assignment table contains the following line:

```
disk sd7 at scsibus1 target 3 lun 0 # HP optical disk drive
```

the SCSI disk number is 7 and the following commands make the device files:

```
cd /dev
MAKEDEV sd7
```

Examples of Device Files

If the desired optical disk drive is on SCSI bus 1 at SCSI ID 3, and the kernel configuration file contains the following line:

```
disk sd7 at scsibus1 target 3 lun 0 # HP optical disk drive
```

then the *sd_number* is 7 and the device file paths you enter are as follows:

`/dev/rsd7a` (volume header)
`/dev/rsd7g` (character device)

Setting the Optical Drive Type in Nonvolatile Memory

Note: If you have not already done so, verify that your kernel SCSI ID mapping table has the appropriate sd (SCSI disk) entries for the optical disk drives. See “Changing SCSI ID Mapping in Kernel Configuration” on page 8 for details.

To use HP optical disk drives, the system must recognize the optical drives as disk drives (using the sd driver) at system boot time. If you are adding Hewlett-Packard 1.2 gigabyte or equivalent model magneto-optical disk drives, the system may not recognize these as disk drives, and thus cannot write to or read from them.

The following steps explain how to correct this condition:

1. Install the SG loadable driver if it is not already installed. See “Configuring SCSI Robotic Controls” on page 2 for information on how to install the SG loadable driver.

Note: Usually Media Manager uses the SG driver to access robotic controls. In the following step, the `/dev` path must allow Media Manager to access the optical disk drive through the SG driver. Be sure to specify the SCSI ID for the optical disk drive, *not* the SCSI ID for the robotic control.

2. Use `/usr/openv/volmgr/bin/scsi_command` to change the optical drive's device type (stored in the drive's nonvolatile memory) from *optical memory* to *disk*. The format of this command is as follows:

   `scsi_command -d /dev/sg/c`*`controller`*`t`*`id`*`d0s0 -disk`

   Where:

   controller is the number of the SCSI controller

   id is the SCSI ID of the disk drive

   For example, if the Hewlett-Packard 1.2 gigabyte magneto-optical disk drive is on controller 1 at SCSI ID 3, enter the following command:

   ```
   scsi_command -d /dev/sg/c1t3d0s0 -disk
   ```

3. Reboot the system to allow the drive to be recognized as a disk drive during system initialization by the kernel's SCSI disk (sd) driver. If you have done kernel reconfiguration, ensure the kernel is in place prior to reboot.

Rebuilding a SunOS Kernel

Read this topic if you have modified the SCSI ID mapping in the kernel configuration table or have added a new drive type to the kernel source by altering the `st_conf.c` or `stdef.h` files.

After you have completed modifications to the SunOS kernel configuration for the type or number of drives, as described in previous topics, you are ready to reconfigure and rebuild the kernel.

This procedure is explained in the following steps:

1. Determine the name of your kernel by using one of the following commands.

   ```
   cat /etc/motd
   ```

   ```
   SunOS Release 4.1.3(MY_KERN) #19:Tue Feb 15 09:55:41 CST 1994
   ```

```
strings /vmunix | grep SunOS
SunOS Release 4.1.3(MY_KERN) #19:Tue Feb 15 09:55:41 CST 1994
```

In these examples, the name of the running kernel is most likely MY_KERN.

2. Use the `arch` command to determine the kernel architecture.

   ```
   /usr/bin/arch -k
   sun4m
   ```

3. Use the `config` utility on the kernel configuration file as follows:

   a. Change your working directory as appropriate:

      `cd /sys/`*`arch`*`/conf`

      Where *arch* is the kernel architecture value obtained in step 2. For example:

      ```
      cd /sys/sun4m/conf
      ```

   b. Run the utility on the configuration file:

      `/etc/config` *kernel_name*

      Where *kernel_name* is the value obtained in step 1 or a new name for your kernel. For example:

      ```
      /etc/config MY_KERN
      ```

4. Build the new kernel using `make` in the appropriate directory:

   `cd ../`*`kernel_name`*

   ```
   make
   ```

   Where *kernel_name* is the value used in step 3.

   This results in a new file named `vmunix` created in your current working directory.

5. Before booting with the new kernel created in the previous step, do the following:

   a. Ensure there is enough disk space in the / partition, then make a copy of the old kernel:

```
cp /vmunix /vmunix.old
```

b. Replace the old kernel with the new one.

```
cp vmunix /vmunix
```

6. Reboot the system.

For more detailed information, see the `st(4s)`, `sd(4s)`, and `config(8)` man pages.

Command Summary

The following is a summary of commands that may be useful when configuring devices. See the procedures in this chapter for examples of their usage.

`/usr/etc/modstat`
: Shows the loadable drivers that are currently loaded.

`/usr/openv/volmgr/bin/driver/sg.install`
: Loads the SG driver.

`/usr/etc/modunload -id` *n*
: Unloads the loadable driver that has Id *n*, as shown by modstat.

`cat /etc/motd`
: Displays the name of the kernel configuration file.

`arch -k`
: Displays the kernel architecture.

`MAKEDEV st` *st_number*
: Creates SCSI tape device files, where *st_number* is the tape device number configured to the desired SCSI bus and SCSI ID in the kernel configuration file.

`dmesg`
: Shows the vendor and product strings for the drives on your system, when it is executed shortly after a boot.

`/etc/config` *kernel_name*

Builds system configuration files prior to rebuilding the kernel. *kernel_name* is the name of the kernel configuration file as returned by `cat /etc/motd` (for example, GENERIC).

`make`

Creates a new kernel file named `vmunix` in your current working directory. This working directory should be `/etc/config/`*kernel_name*, where *kernel_name* is the name of the kernel configuration file as returned by `cat /etc/motd` (for example, GENERIC).

`/usr/openv/volmgr/bin/sgscan`

Allows you to determine the SCSI devices connected to a Auspex SunOS server by executing a SCSI inquiry on all device files in `/dev/sg`.

`/usr/openv/volmgr/bin/spscan`

Allows you to determine the SCSI devices connected to an Auspex Storage Processor by executing a SCSI inquiry on all `/dev/asc*` device files.

`/usr/openv/volmgr/bin/vmconf`

Provided with Media Manager, this script eases device setup in less complex configurations.

NetBackup 3.2 Media Manager Device Configuration Guide for UNIX

Sun4/SPARC Running Solaris 2.5/2.5.1/2.6/7.0

This chapter explains how to configure devices for use with Media Manager on a Sun4/SPARC platform. You configure drives and robots using one of the available Media Manager administrative interfaces.

The major topics included are as follows:

- Installing SCSI Pass-Through Drivers
- Configuring Robotic Controls
- Configuring Tape Drives
- Configuring HP Optical Disk Drives
- Command Summary

To determine which drive and robot types are supported, refer to the *NetBackup Release Notes - UNIX*.

Observe the following points when performing the configurations described in this chapter:

- When configuring devices, you should attach all peripherals and reboot the system with the reconfigure option (`boot -r`).
- When removing or replacing adapter cards, you should remove all device files previously associated with the adapter card.
- If you use the Automated Cartridge System (ACS) robotic software, you must ensure that the SunOS/BSD Source Compatibility Package is installed, so that ACS software can make use of shared libraries in `/usr/ucblib`.

- Always install the sg driver (see “Installing SCSI Pass-Through Drivers” on page 29). This driver is used by `avrd` (automatic volume recognition daemon) to scan drives, in addition to being required for communicating with SCSI devices and also for locate-block tape positioning.
- VERITAS does not recommend or support the use of single-ended to differential SCSI converters on Media Manager controlled devices. You may encounter problems if you use these converters.
- You can copy code examples from the on-line version of this manual, located in `/usr/openv/volmgr/dev_config_guide.txt`. This file is installed with Media Manager software.

Installing SCSI Pass-Through Drivers

Media Manager software provides its own driver for communicating with SCSI-controlled robotic peripherals. This driver is called the SCSA (Generic SCSI pass-through driver), also referred to as the sg driver.

The sg driver is also used

- To set the optical drive type (as explained in “Setting the HP Optical Drive Type in Nonvolatile Memory” on page 49).
- By `avrd` for scanning drives.
- By NetBackup for locate-block positioning.

The following procedure explains how to load the sg driver. Perform these steps as the root user.

1. Determine if an sg driver is loaded by using the following command:

   ```
   /usr/sbin/modinfo | grep sg
   ```

   ```
   141 fc580000  2d8c 116   1  sg (SCSA Generic Revision: 3.2)
   153 fc7fa000  1684  49   1  msgsys (System V message facility)
   ```

   In this example, there already is an `sg` driver loaded, so you must remove it before proceeding.

2. Remove the existing driver by executing:

   ```
   /usr/sbin/rem_drv sg
   ```

3. Install the new sg driver, with the following command:

   ```
   /usr/openv/volmgr/bin/driver/sg.install
   ```

   Once the driver has been installed, it is not necessary to reboot the system, or run the `sg.install` command during or after each system boot.

   To uninstall the `sg` driver, at a later time, use the following command:

   ```
   /usr/sbin/rem_drv sg
   ```

Configuring the SG Driver

This section contains instructions for configuring the SG driver for SCSI targets 8 thru 15 for Fast/Wide Adapter Cards.

1. Execute the `sg.build` script to add targets 8 thru 15 to `/kernel/drv/st.conf`.

   An example of the arguments to the script follows:

```
/usr/openv/volmgr/bin/sg.build st.conf -mt 15 -ml 1 -st
/usr/openv/volmgr/bin/driver/st.conf
```

   `sg.build` will create a file that should be used to replace the following seven entries in the `/kernel/drv/st.conf` file:

```
name="st" class="scsi"
    target=0 lun=0;
   .......
name="st" class="scsi"
    target=6 lun=0;
```

   When you finish changing the `st.conf` file, reboot the system with the reconfigure option (`boot -r`).

2. Execute the `sg.build` script to add targets 8 thru 15 to the `/usr/openv/volmgr/bin/driver/sg.links` file.

CAUTION! The field separator is a tab between the addr=#,#; and the sg/c\N0t#l# field.

   An example of the arguments to the script and the file that is generated follow:

```
/usr/openv/volmgr/bin/sg.build sg.links -mt 15 -ml 1 -sl
/usr/openv/volmgr/bin/driver/sg.links
```

```
type=ddi_pseudo;name=sg;addr=0,0;       sg/c\N0t0l0
type=ddi_pseudo;name=sg;addr=1,0;       sg/c\N0t1l0
type=ddi_pseudo;name=sg;addr=2,0;       sg/c\N0t2l0
type=ddi_pseudo;name=sg;addr=3,0;       sg/c\N0t3l0
type=ddi_pseudo;name=sg;addr=4,0;       sg/c\N0t4l0
type=ddi_pseudo;name=sg;addr=5,0;       sg/c\N0t5l0
type=ddi_pseudo;name=sg;addr=6,0;       sg/c\N0t6l0
type=ddi_pseudo;name=sg;addr=8,0;       sg/c\N0t8l0
type=ddi_pseudo;name=sg;addr=9,0;       sg/c\N0t9l0
type=ddi_pseudo;name=sg;addr=a,0;       sg/c\N0t10l0
type=ddi_pseudo;name=sg;addr=b,0;       sg/c\N0t11l0
type=ddi_pseudo;name=sg;addr=c,0;       sg/c\N0t12l0
```

```
type=ddi_pseudo;name=sg;addr=d,0;       sg/c\N0t13l0
type=ddi_pseudo;name=sg;addr=e,0;       sg/c\N0t14l0
type=ddi_pseudo;name=sg;addr=f,0;       sg/c\N0t15l0
type=ddi_pseudo;name=sg;addr=0,1;       sg/c\N0t0l1
type=ddi_pseudo;name=sg;addr=1,1;       sg/c\N0t1l1
type=ddi_pseudo;name=sg;addr=2,1;       sg/c\N0t2l1
type=ddi_pseudo;name=sg;addr=3,1;       sg/c\N0t3l1
type=ddi_pseudo;name=sg;addr=4,1;       sg/c\N0t4l1
type=ddi_pseudo;name=sg;addr=5,1;       sg/c\N0t5l1
type=ddi_pseudo;name=sg;addr=6,1;       sg/c\N0t6l1
type=ddi_pseudo;name=sg;addr=8,1;       sg/c\N0t8l1
type=ddi_pseudo;name=sg;addr=9,1;       sg/c\N0t9l1
type=ddi_pseudo;name=sg;addr=a,1;       sg/c\N0t10l1
type=ddi_pseudo;name=sg;addr=b,1;       sg/c\N0t11l1
type=ddi_pseudo;name=sg;addr=c,1;       sg/c\N0t12l1
type=ddi_pseudo;name=sg;addr=d,1;       sg/c\N0t13l1
type=ddi_pseudo;name=sg;addr=e,1;       sg/c\N0t14l1
type=ddi_pseudo;name=sg;addr=f,1;       sg/c\N0t15l1
```

3. Execute the `sg.build` script to add targets 8 thru 15 to the `/usr/openv/volmgr/bin/driver/sg.conf` file.

   An example of the arguments to the script and the file generated follow:

```
/usr/openv/volmgr/bin/sg.build sg.conf -mt 15 -ml 1 -sc
/usr/openv/volmgr/bin/driver/sg.conf
```

```
name="sg" class="scsi" target=0 lun=0;
name="sg" class="scsi" target=0 lun=1;
name="sg" class="scsi" target=1 lun=0;
name="sg" class="scsi" target=1 lun=1;
name="sg" class="scsi" target=2 lun=0;
name="sg" class="scsi" target=2 lun=1;
name="sg" class="scsi" target=3 lun=0;
name="sg" class="scsi" target=3 lun=1;
name="sg" class="scsi" target=4 lun=0;
name="sg" class="scsi" target=4 lun=1;
name="sg" class="scsi" target=5 lun=0;
name="sg" class="scsi" target=5 lun=1;
name="sg" class="scsi" target=6 lun=0;
name="sg" class="scsi" target=6 lun=1;
name="sg" class="scsi" target=8 lun=0;
name="sg" class="scsi" target=8 lun=1;
name="sg" class="scsi" target=9 lun=0;
name="sg" class="scsi" target=9 lun=1;
name="sg" class="scsi" target=10 lun=0;
name="sg" class="scsi" target=10 lun=1;
name="sg" class="scsi" target=11 lun=0;
name="sg" class="scsi" target=11 lun=1;
name="sg" class="scsi" target=12 lun=0;
name="sg" class="scsi" target=12 lun=1;
name="sg" class="scsi" target=13 lun=0;
name="sg" class="scsi" target=13 lun=1;
name="sg" class="scsi" target=14 lun=0;
name="sg" class="scsi" target=14 lun=1;
name="sg" class="scsi" target=15 lun=0;
name="sg" class="scsi" target=15 lun=1;
```

4. Edit the file `/etc/devlink.tab` and remove the devices for the SG driver.
5. Uninstall the SG driver (`/usr/sbin/rem_drv sg`).
6. Install the new SG driver (`/usr/openv/volmgr/bin/driver/sg.install`).

Configuring Robotic Controls

Robots are controlled through a SCSI, an RS-232, or a network connection. Configuration for network controlled robotic libraries is discussed in the appendices of the *Media Manager System Administrator's Guide - UNIX*. SCSI and RS-232 control is covered in the following sections.

Configuring SCSI Robotic Controls

Read this topic if you plan to use a robotic storage device that is controlled through a SCSI robotic connection. Supported SCSI robots include:

- ODL - Optical Disk Library
- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TL4 - Tape Library 4MM
- TL8 - Tape Library 8MM
- TLD - Tape Library DLT
- TS8 - Tape Stacker 8MM
- TSD - Tape Stacker DLT
- TSH - Tape Stacker Half-inch

See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with the above robot types.

When communicating with SCSI-controlled robotic peripherals, Media Manager software utilizes the SCSA Generic (sg) driver. This driver is provided with the NetBackup robotic software and you must install it before continuing with the instructions in this topic (see "Installing SCSI Pass-Through Drivers" on page 29 for details).

To display the device files that are available to be used through the sg driver, use the `sgscan` command with the `all` parameter, and note the lines that indicate the Changer devices, as in the following example:

```
# /usr/openv/volmgr/bin/sgscan all

/dev/sg/c0t5l0: Tape (/dev/rmt/0): "HP      C1537A"
/dev/sg/c0t6l0: Cdrom: "TOSHIBA XM-5401TASUN4XCD"
/dev/sg/c1t2l0: Tape (/dev/rmt/7): "EXABYTE EXB-85058HE-0000"
/dev/sg/c1t4l0: Tape (/dev/rmt/9): "EXABYTE EXB-8900MH000202"
/dev/sg/c1t5l0: Changer: "EXABYTE EXB-210"
/dev/sg/c2t2l0: Tape (/dev/rmt/10): "Quantum DLT4000"
/dev/sg/c2t5l0: Tape (/dev/rmt/11): "QUANTUM DLT7000"
/dev/sg/c3t0l0: Disk (/dev/rdsk/c1t0d0): "FUJITSU M2952ESP SUN2.1G"
/dev/sg/c3t3l0: Disk (/dev/rdsk/c1t3d0): "FUJITSU M2952ESP SUN2.1G"
/dev/sg/c4t4l0: Tape (/dev/rmt/4): "Quantum DLT4000"
/dev/sg/c4t5l0: Tape (/dev/rmt/5): "Quantum DLT4000"
/dev/sg/c5t0l0: Disk (/dev/rdsk/c5t0d0): "SONY    SMO-F541"
/dev/sg/c5t1l0: Disk (/dev/rdsk/c5t1d0): "SONY    SMO-F541"
/dev/sg/c5t2l0: Disk (/dev/rdsk/c5t2d0): "SEAGATE ST11200N SUN1.05"
/dev/sg/c5t6l0: Disk (/dev/rdsk/c5t6d0): "SEAGATE ST11200N SUN1.05"
/dev/sg/c6t3l0: Changer: "SONY    DMS-B35"
/dev/sg/c6t5l0: Tape (/dev/rmt/6): "SONY    GY-2120"
/dev/sg/c7t0l0: Disk (/dev/rdsk/c7t0d0): "SEAGATE ST32550W SUN2.1G"
/dev/sg/c7t3l0: Disk (/dev/rdsk/c7t3d0): "MICROP  4221-09   1128RA"
/dev/sg/c7t4l0: Disk (/dev/rdsk/c7t4d0): "MICROP  4221-09MZ  Q4D"
/dev/sg/c8t2l0: Tape (/dev/rmt/14): "Quantum DLT4000"
/dev/sg/c8t3l0: Changer: "STK     9740"
/dev/sg/c8t4l0: Tape (/dev/rmt/13): "STK     SD-3"
/dev/sg/c8t6l0: Changer: "STK     9710"
/dev/sg/c9t0l0: Changer: "EXABYTE Exabyte 18D"
/dev/sg/c9t1l0: Tape (/dev/rmt/15): "Quantum DLT4000"
```

Note: Specific device types can be filtered from the output using other forms of `sgscan`.

Usage: `sgscan [all|basic|changer|disk|tape] [conf] [-v]`

Examples of SCSI Robotic Control Device Files

Example 1

Using the above `sgscan` output, if the SCSI robotic control for an Exabyte 210 is connected to SCSI ID 5 of adapter 1, you use the following path:

```
/dev/sg/c1t5l0
```

Example 2

Using the above `sgscan` output, if the SCSI robotic control for a Sony Optical Disk Library is connected to SCSI ID 3 of adapter 6, you use the following path:

```
/dev/sg/c6t3l0
```

Example 3

Using the above `sgscan` output, if the SCSI robotic control for an STK 9710 is connected to SCSI ID 6 of adapter 8 and you want to use TLD robotics, you use the following path:

```
/dev/sg/c8t6l0
```

Example 4

If the SCSI robotic control for a DLT2700, DLT4700, or HP C1560B was connected to SCSI ID 5 of adapter 0, you use the following path:

```
/dev/sg/c0t5l1
```

Note that logical unit number 1 is used for those devices. The sg driver configuration can be modified so `sgscan` lists LUN 1 devices. In the `sgscan` output shown above, the configuration was not modified.

Example 5

Using the above `sgscan` output, even if the SCSI robotic control for an STK 9740 is connected to SCSI ID 3 of adapter 8, you would not enter any path to configure ACS robotic control. Instead, assuming ACS control over the network, enter the appropriate ACSLS Host name. (If you want to use TLD robotics to control the 9740, specify the path `/dev/sg/c8t3l0`)

Example 6 (IBM 3570 B-series Stackers)

If there is one drive in the stacker, the robotic control is LUN 1 of the drive's SCSI ID. If there are two drives in the stacker, the robotic control is LUN 1 of the Drive 1 SCSI ID. The SCSI ID's are setable and viewable using the front panel on the stacker.

The robotic control for the IBM 3570 B01 / B02 is TLD, so if there are two drives, they may be connected to different host systems. If this is the case, the host system which is connected to drive 1 must also have the robotic control. Also, the library should be in RANDOM mode and BASE configuration. See the Operator's Guide supplied with the unit for information on setting library mode and configuration.

Assume a configuration as follows:

```
# /usr/openv/volmgr/bin/sgscan
```

```
/dev/sg/c0t0l0: Disk (/dev/rdsk/c0t0d0): "IBM     DCAS32160SUN2.1G"
/dev/sg/c0t6l0: Cdrom: "TOSHIBA XM5701TASUN12XCD"
/dev/sg/c1t5l0: Tape (/dev/rmt/1): "IBM     03570B02"
/dev/sg/c1t6l0: Tape (/dev/rmt/2): "IBM     03570B02"
#
```

If drive 1 is SCSI ID 5, the robotic control for the stacker is `/dev/sg/c1t5l1`.

Configuring RS-232 Robotic Controls

Read this topic if you plan to use a robotic device that is controlled through an RS-232 connection. Supported RS-232 robots include the following:

- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TLD - Tape Library DLT

See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with the above robot types.

Configuring RS-232 controlled devices normally consists of choosing between the A and B serial console ports.

Examples of RS-232 Robotic Control Device Files

Typical path names to use when adding a RS-232 controlled robotic device to the configuration are as follows:

```
serial port A:  /dev/term/a
serial port B:  /dev/term/b
```

Configuring Tape Drives

Using Berkeley-Style Close

The examples in this section show Berkeley-style close for tape drives as indicated by the letter `b` after the density specification. You must specify Berkeley-style close for tape devices that you configure under Media Manager.

The terms *Berkeley-style close* and *AT&T style close* refer to where a tape is left logically positioned after a close operation (in relation to a tape mark). One style leaves an application logically positioned before a tape mark and the other leaves it after. Applications must assume where the tape is left after a close in order to establish the correct orientation the next time they do a tape-position or read operation. Some operating systems allow tape devices to be configured with either type of close. NetBackup assumes it is using Berkeley-style close.

Fast-Tape Positioning (locate-block)

For AIT, DLT, Exabyte, DTF, and half-inch tape drives, Media Manager supports the SCSI `locate-block` command for positioning to a specific block on a tape. This approach improves tape-positioning times over the alternative, which is the forward-space-file / record method.

Enabling locate-block

NetBackup and Storage Migrator use the `locate-block` command by default if you did not uninstall the sg passthru driver as explained in “Installing SCSI Pass-Through Drivers” on page 29. The driver is automatically installed with Media Manager.

Disabling locate-block

To disable locate-block positioning, execute the following:

```
touch /usr/openv/volmgr/database/NO_LOCATEBLOCK
```

With locate-block positioning disabled, NetBackup uses the forward-space-file / record method and Storage Migrator skips file marks.

No Rewind Device Files

When adding tape drives to a Media Manager configuration, you need only specify a no rewind on close device path. To display the tape device files that are configured on your system, use the sgscan command with the tape parameter.

```
# /usr/openv/volmgr/bin/sgscan tape

/dev/sg/c0t5l0: (/dev/rmt/0): "HP      C1537A"
/dev/sg/c1t2l0: (/dev/rmt/7): "EXABYTE EXB-85058HE-0000"
/dev/sg/c1t4l0: (/dev/rmt/9): "EXABYTE EXB-8900MH000202"
/dev/sg/c2t2l0: (/dev/rmt/10): "Quantum DLT4000"
/dev/sg/c2t5l0: (/dev/rmt/11): "QUANTUM DLT7000"
/dev/sg/c4t4l0: (/dev/rmt/4): "Quantum DLT4000"
/dev/sg/c4t5l0: (/dev/rmt/5): "Quantum DLT4000"
/dev/sg/c6t5l0: (/dev/rmt/6): "SONY    GY-2120"
/dev/sg/c8t2l0: (/dev/rmt/14): "Quantum DLT4000"
/dev/sg/c8t4l0: (/dev/rmt/13): "STK     SD-3"
/dev/sg/c9t1l0: (/dev/rmt/15): "Quantum DLT4000"
```

Note: All device types can be displayed in the output using the all parameter with sgscan. This command can be helpful for associating tape devices with other SCSI devices that may be configured on the same adapter.

Usage: `sgscan [all|basic|changer|disk|tape] [conf] [-v]`

No rewind on close device files are in the /dev/rmt directory, and have the following format:

/dev/rmt/*Logical_drive*cbn

Where:

Logical_drive is the logical drive id, as shown by the sgscan command.

The c indicates compression.

The b indicates Berkeley-style close.

The n indicates no rewind on close.

Examples of No Rewind Device Files

Example 1

Using the sgscan output, if an Exabyte 8505C drive is connected to SCSI ID 2 of adapter 1, the device path you use follows:

```
/dev/rmt/7cbn
```

Example 2

Using the `sgscan` output, if a DLT7000 drive is connected to SCSI ID 5 of adapter 2, the device path you use follows:

```
/dev/rmt/11cbn
```

Configuring Nonstandard Tape Drives

Read this topic if you plan to use any of the following drive types. These are nonstandard drive types requiring changes to the kernel before you can use them.

- Exabyte (models 8500, 8505, 8505XL, 8500C, or 8900)
- Fujitsu M2488
- HP 4-mm DAT
- IBM 3570 and 3590
- Quantum DLT2000, DLT4000, or DLT7000
- Sony AIT and DTF
- STK half-inch cartridge
- Tandberg QIC and QIC 150

CAUTION! As shown by the `st.conf` examples in this section, you must configure non-QIC tape drives as variable-mode devices if they are to be used by Media Manager on Solaris platforms. Otherwise, NetBackup is able to write data, but not read it. During a read, you see a “not in tar format” error. The terms *variable mode* or *fixed mode* refers to the behavior of reads and writes and the way the kernel packs physical tape records into logical tape records for an application. Variable-mode devices allow more flexibility in reading previously written tapes. Many tape devices can be accessed in either mode. NetBackup assumes variable mode for non-QIC drives.

Note on Case and Spaces in `st.conf` Entries

Upper and lower case are significant. For example, using Hp instead of HP would not work.

Spaces are significant within quoted strings in the `/kernel/drv/st.conf` file. The area that users most frequently have trouble with is the vendor field, which must always be eight characters in length. For example, if the vendor / product string for an HP C1533A drive is as follows:

`"HP    C1533A"` (HP*ssssss* is the vendor field)

If you were to omit some of the spaces in the vendor field, such as in:

`"HP C1533A"` (HP*ss* is now the vendor field) the drive would not be recognized correctly.

The best way to ensure that your entries are accurate is to copy them, whenever possible, from `dev_config_guide.txt`.

Additions to `st.conf`

An entry must be included for the drive type you are running. The changes in this section were tested and are known to work. Other settings may also work.

CAUTION! Pay special attention to the second half of this list, where the third parameter must be 0, (indicating variable mode). Not making the following change will cause restores to fail and may result in data loss.

```
tape-config-list =

"EXABYTE  EXB8500C", "Exabyte EXB-8500C 8mm Helical Scan", "EXB-8500C”,
"EXABYTE  EXB-8505", "Exabyte EXB-8505 8mm Helical Scan", "EXB-8505",
"EXABYTE  EXB-8500", "Exabyte EXB-8500 8mm Helical Scan", "EXB-8500",
"EXABYTE  EXB-8900", "Exabyte EXB-8900 Mammoth", "EXB-8900",
"FUJITSU  M2488", "Fujitsu M2488", "FJ-D3",
"HP       HP354", "HP 4mm DAT Drive", "HP-DAT",
"HP       C1533A", "HP DAT Autoloader", "HP-DAT",
"HP       C1557A", "HP Dat DDS3 Autoloader", "HP-DAT-DDS3",
"IBM      03590", "IBM 3590 1/2 Inch Cartridge", "IBM-3590",
"IBM      03570", "IBM 3570 1/2 Inch Cartridge", "IBM-3590",
"Metrum   RSP-2150", "Metrum VHS Drive", "Metrum",
"ARCHIVE VIPER 150", "Archive 150 Tape", "ARCHIVE_VIP",
"TANDBERG SLR5 4/8GB",  "Tandberg 8 Gig QIC",    "TAND-8G-VAR",
"SONY     GY-2120", "Sony DTF Drive", "gy20-data",
"SONY     SDX-300C", "SONY 8mm AIT", "SONY_AIT",
"DEC      DLT2000", "DEC DLT Tape Drive", "DEC-DLT",
```

```
"DEC     DLT2700", "DEC DLT Tape Stacker", "DEC-DLT",
"Quantum DLT2000", "Quantum DLT Tape Drive", "DEC-DLT",
"Quantum DLT4000", "Quantum DLT Tape Drive", "DEC-DLT",
"Quantum DLT4700", "Quantum DLT Tape Stacker", "DEC-DLT",
"QUANTUM DLT7000", "Quantum DLT7000 Tape Drive", "Q-DLT7000",
"Quantum DLT2700", "Quantum DLT Tape Stacker", "DEC-DLT",
"STK     4781", "STK 1/2 Inch Cartridge (4480)", "STK-4781",
"STK     4791", "STK 1/2 Inch Cartridge (Silverton)", "STK-4791",
"STK     4890", "STK 1/2 Inch Cartridge (Twin Peaks)", "STK-4890",
"STK     9840", "STK 1/2 Inch Cartridge (9840)", "STK-9840",
"STK     SD-3", "STK 1/2 Inch Cartridge (Redwood)", "STK-SD-3";


EXB-8500C = 1,0x35,0,0x9639,4,0x14,0x15,0x8C,0x00,3;
EXB-8505 = 1,0x35,0,0x9639,4,0x14,0x15,0x8C,0x00,3;
EXB-8500 = 1,0x35,0,0x9639,4,0x14,0x00,0x00,0x15,2;
EXB-8900 = 1,0x35,0,0x9639,4,0x27,0x27,0x27,0x00,3;
FJ-D3 = 1,0x21,0,0xCA19,4,0x09,0x09,0x09,0x09,0;
HP-DAT = 1,0x34,0,0x9639,4,0x0,0x0,0x0,0x0,3;
HP-DAT-DDS3 = 1,0x34,0,0,0x9639,4,0x0,0x8c,0x8c,0x8c,3;
IBM-3590 = 1,0x24,0,0x1c63d,4,0x0,0x0,0x0,0x0,3;
Metrum = 1,0x36,0,0x9639,4,0xf0,0xf0,0xf0,0xf0,3;
ARCHIVE_VIP = 1,0x32,512,0x163a,4,0x0,0x0,0x0,0x0,3;
TAND-8G-VAR = 1,0x37,0,0x963b,4,0xa0,0xd0,0xd0,0xd0,3;
gy20-data = 1,0x36,0,0xd659,1,0x00,0;
DEC-DLT = 1,0x36,0,0x9639,4,0x0,0x0,0x0,0x0,3;
Q-DLT7000 = 1,0x36,0,0x9639,4,0x84,0x83,0x0,0x85,3;
SONY_AIT = 1,0x34,0,0x9639,4,0x13,0x0,0x8C,0x8C,3;
STK-4781 = 1,0x24,0,0xld43d,1,0x00,0;
STK-4791 = 1,0x24,0,0x1d67d,1,0x00,0;
STK-4890 = 1,0x24,0,0x1d67d,1,0x00,0;
STK-9840 = 1,0x36,0,0x1d639,1,0x00,0;
STK-SD-3 = 1,0x24,0,0x1d67d,1,0x00,0;
```

CAUTION! Reboot the system when you are done changing the kernel, using the reconfigure option (`boot -r`) to allow the kernel's SCSI tape (st) driver to recognize the drives as the correct type during system initialization.

Adding Logical Unit Number Entries

If the devices you are adding utilize the logical unit number (LUN) concept, (such as a half-inch cartridge drives that attach to an STK Automated Cartridge System) you must also add entries of the following form to the `st.conf` file:

```
name="st" class="scsi"
  target=SCSI_ID lun=LUN
```

Where:

SCSI_ID is the SCSI ID to which the drives are attached.

LUN is the logical unit number for the drive.

For example, if a drive is configured at logical unit number 4 of SCSI ID 3, you add the following entry to `st.conf`:

```
name="st" class="scsi"
  target=3 lun=4
```

You can use the `sg.build` script to generate entries for the `st.conf` file.

Multiple Logical Unit Numbers per SCSI Target

This section describes how to install a device with multiple Logical Unit Numbers per SCSI target. This is necessary in a fibre channel environment.

1. Execute the `sg.build` script to add targets 8 thru 15 to `/kernel/drv/st.conf`.

   An example of the arguments to the script follows:

```
/usr/openv/volmgr/bin/sg.build st.conf -mt 15 -ml 1 -st
/usr/openv/volmgr/bin/driver/st.conf
```

   `sg.build` will create a file that should be used to replace the following seven entries in the `/kernel/drv/st.conf` file:

```
name="st" class="scsi"
    target=0 lun=0;
   .......
name="st" class="scsi"
    target=6 lun=0;
```

2. Execute the `sg.build` script to add targets 8 thru 15 to the `/usr/openv/volmgr/bin/driver/sg.conf` file.

   An example of the arguments to the script and the file generated follow:

```
/usr/openv/volmgr/bin/sg.build sg.conf -mt 15 -ml 1 -sc
/usr/openv/volmgr/bin/driver/sg.conf
```

```
name="sg" class="scsi" target=0 lun=0;
name="sg" class="scsi" target=0 lun=1;
name="sg" class="scsi" target=1 lun=0;
name="sg" class="scsi" target=1 lun=1;
name="sg" class="scsi" target=2 lun=0;
name="sg" class="scsi" target=2 lun=1;
name="sg" class="scsi" target=3 lun=0;
name="sg" class="scsi" target=3 lun=1;
name="sg" class="scsi" target=4 lun=0;
name="sg" class="scsi" target=4 lun=1;
name="sg" class="scsi" target=5 lun=0;
name="sg" class="scsi" target=5 lun=1;
```

```
name="sg" class="scsi" target=6 lun=0;
name="sg" class="scsi" target=6 lun=1;
name="sg" class="scsi" target=8 lun=0;
name="sg" class="scsi" target=8 lun=1;
name="sg" class="scsi" target=9 lun=0;
name="sg" class="scsi" target=9 lun=1;
name="sg" class="scsi" target=10 lun=0;
name="sg" class="scsi" target=10 lun=1;
name="sg" class="scsi" target=11 lun=0;
name="sg" class="scsi" target=11 lun=1;
name="sg" class="scsi" target=12 lun=0;
name="sg" class="scsi" target=12 lun=1;
name="sg" class="scsi" target=13 lun=0;
name="sg" class="scsi" target=13 lun=1;
name="sg" class="scsi" target=14 lun=0;
name="sg" class="scsi" target=14 lun=1;
name="sg" class="scsi" target=15 lun=0;
name="sg" class="scsi" target=15 lun=1;
```

3. Execute the `sg.build` script to add targets 8 thru 15 to the `/usr/openv/volmgr/bin/driver/sg.links` file.

CAUTION! The field separator is a tab between the addr=#,#; and the sg/c\N0t#l# field.

An example of the arguments to the script and the file that is generated follow:

```
/usr/openv/volmgr/bin/sg.build sg.links -mt 15 -ml 1 -sl
/usr/openv/volmgr/bin/driver/sg.links
```

```
type=ddi_pseudo;name=sg;addr=0,0;       sg/c\N0t0l0
type=ddi_pseudo;name=sg;addr=1,0;       sg/c\N0t1l0
.
.
type=ddi_pseudo;name=sg;addr=3,1;       sg/c\N0t3l1
type=ddi_pseudo;name=sg;addr=4,1;       sg/c\N0t4l1
type=ddi_pseudo;name=sg;addr=5,1;       sg/c\N0t5l1
type=ddi_pseudo;name=sg;addr=6,1;       sg/c\N0t6l1
type=ddi_pseudo;name=sg;addr=1,2;       sg/c\N0t1l2 [1]
```

4. Remove the zero preceding the last * in the for statement, to get all LUNs to print.

```
for i in /dev/sg/c*t*[dl]0*; do
for i in /dev/sg/c*t*[dl]*; do
```

5. Install or re-install the sg driver `./install.sg`.

1. add the lun # in both places

Refer to “Installing SCSI Pass-Through Drivers” on page 29 for more details.

6. Run the `sgscan` command to display the `/dev/sg/`*device numbers* and SCSI inquiry strings

7. If you are installing an optical library, run `scsi_command -d /dev/sg/`*sg_id* `-inquiry` to verify each optical target.

   Then run `scsi_command -d /dev/sg/`*sg_id* `-disk` to change the eeprom for the optical drive from optical to disk, so the Solaris sd driver can communicate with the drive.

8. Edit the `/kernel/drv/sd.conf` file.

```
#
# Copyright (c) 1992, by Sun Microsystems, Inc.
#
#ident  "@(#)sd.conf    1.8     93/05/03 SMI"

name="sd" class="scsi"
       target=0 lun=0;
name="sd" class="scsi"
       target=1 lun=0;
name="sd" class="scsi"
       target=2 lun=0;
name="sd" class="scsi"
       target=3 lun=0;[2]
name="sd" class="scsi"
       target=4 lun=0;
name="sd" class="scsi"
       target=4 lun=1;[3]
name="sd" class="scsi"
       target=4 lun=2;
name="sd" class="scsi"
       target=4 lun=3;
.
.
.
```

9. Reboot the system using the -r option to reconfigure the devices and you should see the sd driver address the drives during the boot process. (The `sgscan all` command displays them.)

2. Copy the target entry lines that you need

3. Substitute the new lun # as needed

Adding HP 4-mm Drives and HP DAT Autoloaders

Read this section if you plan to use Hewlett-Packard (HP) 4-mm DAT tape drives or HP C1560B DAT Autoloaders.

Note: Other combinations may work, but these settings were functional during testing with an HP35480 drive and HP C1560B Autoloader.

Use the following hardware (tape drive) switch settings on HP35480 4-mm (DAT) drives:

On=1, Off=0

Switch	Setting
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1

Use the following settings on HP C1533A drives in an HP C1560B DAT Autoloader:

Switch	Setting
1	1
2	1
3	1
4	1
5	1
6	1
7	0
8	0

Adding Sony AIT Drives

Read this section if you plan to use Sony AIT tape drives in your configuration.

No Rewind Device Files

When adding tape drives to a Media Manager configuration, you need only specify a no rewind on close device path. To display the no rewind device files that are configured on your system, use the `sgscan` command with the `tape` parameter.

```
# /usr/openv/volmgr/bin/sgscan tape

/dev/sg/c2t5l0: Tape (/dev/rmt/6): "SONY SDX-300C"
```

Using the `sgscan` output, if the drive is connected to SCSI ID 5 of adapter 2, the device path you use follows:

```
/dev/rmt/6cbn
```

Dip Switch Settings

Sony AIT drives have 8 dip switches located on the bottom of the drive. It is important to set these switches correctly, even if it means taking the drives out of robots and checking them.

Some robots (for example, SpectraLogic) provide a way to set the drive switches from the robot itself. For SpectraLogic robots, it doesn't matter what the drive switches are. The Treefrog (215) robot has a dial in the back to set the appropriate OS. The Bullfrog (10000) robot has a means of setting the OS through the touchscreen.

Depending on the version of the AIT drive, drives are shipped from Sony with one of the following two settings as shown below:

Note: Robot vendors and hardware resellers may change the default drive switch settings.

On=1 and Off=0.

Switch	Setting
1	0
2	0
3	0
4	0
5	0

Switch	Setting
6	0
7	1
8	1

Switch	Setting
1	0
2	0
3	0
4	0
5	1
6	0
7	1
8	0

Switches 1 thru 4 are critical for setting the OS type. Usually, switches 5 - 8 can be left set at the default. For Solaris, use the following switch settings:

Switch	Setting
1	0
2	1
3	0
4	1

You can use the following command to determine the current dip switch settings without removing the drives and checking them:

```
/usr/openv/volmgr/bin/scsi_command -d /dev/sg/c2t5l0 -ait
```

The output is as follows:

```
Physical AIT drive switch setting = 0x0 (Default
configuration)

Logical AIT drive switch setting = 0xa (SUN - SunOS and
Solaris)
```

Configuring HP Optical Disk Drives

To use standalone Hewlett-Packard optical-disk drives, the sg driver must be installed (see “Installing SCSI Pass-Through Drivers” on page 29). The system must also be configured to recognize the optical drives as disk drives at system boot time.

If you are adding Hewlett-Packard 1.2 gigabyte or equivalent model magneto-optical disk drives, the system may not recognize these as disk drives and thus cannot use them. See “Setting the HP Optical Drive Type in Nonvolatile Memory” on page 49 for more information.

Creating Device Files

When adding optical disk drives to a Media Manager configuration, you must specify the following device paths:

- Volume header disk device path (partition 0).
- Character device path (partition 6).

To display the disk device files that are configured on your system, use the `sgscan` command with the `disk` parameter:

```
# /usr/openv/volmgr/bin/sgscan disk

/dev/sg/c0t0l0: (/dev/rdsk/c0t0d0): "IBM     DCAS32160SUN2.1G"
/dev/sg/c0t1l0: (/dev/rdsk/c0t1d0): "HP      C1113F"
/dev/sg/c0t2l0: (/dev/rdsk/c0t2d0): "HP      C1113F"
/dev/sg/c0t5l0: (/dev/rdsk/c0t5d0): "HP      C1160F"
/dev/sg/c1t0l0: (/dev/rdsk/c1t0d0): "SONY    SMO-F541"
/dev/sg/c1t1l0: (/dev/rdsk/c1t1d0): "SONY    SMO-F541"
/dev/sg/c1t2l0: (/dev/rdsk/c1t2d0): "SEAGATE ST11200N SUN1.05"
```

Note: All device types can be displayed using the `all` parameter when executing `sgscan`. This command can be helpful for associating disk devices with other SCSI devices that may be configured on the same adapter.

Usage: `sgscan [all|basic|changer|disk|tape] [conf] [-v]`

Optical disk device files are located in the `/dev` directory, and have the following format:

`/dev/rdsk/c`*Adapter*`t`*Target*`d0s0` (volume header device)
`/dev/rdsk/c`*Adapter*`t`*Target*`d0s6` (character device)

Where:

Adapter is the logical adapter number as shown in the `sgscan` output.

Target is the SCSI ID.

Examples of Optical Disk Device Files

Example 1

Using the above sample `sgscan` output, if the desired optical disk drive connects to SCSI ID 5 of adapter card 0, you would use the following device paths:

`/dev/rdsk/c0t5d0s0` (volume header device)
`/dev/rdsk/c0t5d0s6` (character device)

Example 2

Using the above sample `sgscan` output, if the desired optical disk drive connects to SCSI ID 0 of S bus 1 adapter card 1, you would use the following device paths:

`/dev/rdsk/c1t0d0s0` (volume header device)
`/dev/rdsk/c1t0d0s6` (character device)

Setting the HP Optical Drive Type in Nonvolatile Memory

To use HP optical disk drives, the system must recognize the optical drives as disk drives at system boot time. If you are adding Hewlett-Packard 1.2 gigabyte or equivalent model magneto-optical disk drives, the system may not recognize these as disk drives. The following steps explain how to correct this condition:

1. Install the sg loadable driver if it is not already installed. See “Installing SCSI Pass-Through Drivers” on page 29 for information on how to install this driver.

2. Use the `scsi_command` command to change the device type (stored in the drive’s nonvolatile memory) from optical memory to disk. The format of the command is as follows.

Note: The `/dev` path allows Media Manager to access the optical disk drive through the sg driver. This is an exception to the usual case where Media Manager uses the sg driver to access robotic controls. Therefore, be sure to specify the SCSI ID for the optical disk drive, *not* the SCSI ID for the robotic control.

```
/usr/openv/volmgr/bin/scsi_command -d /dev/sg/sg_id -disk
```

Where *sg_id* is the logical identifier assigned to the optical disk drive for use by the sg driver. See "Configuring SCSI Robotic Controls" on page 33 for information on determining the logical identifier.

3. Reboot the system with the reconfigure option (`boot -r`) to allow the drive to be recognized as a disk drive by the kernel's SCSI disk (sd) driver during system initialization.

Command Summary

The following is a summary of commands that may be useful when configuring devices. See the procedures in this chapter for examples of their usage.

`/usr/sbin/modinfo | grep sg`
: Displays whether or not the sg driver is installed.

`/usr/openv/volmgr/bin/driver/sg.install`
: Installs the sg driver for the first time.

`/usr/sbin/rem_drv sg`
: Uninstalls sg driver.

`/usr/openv/volmgr/bin/sg.build all`
: `-mt` *max_target*
`-ml` *max_lun*
`-st` *st.conf_file*
`-sc` *sc.conf_file*
`-sl` *sg.links_file*
Updates `st.conf`, `sg.conf`, and `sg.links`, and generates wide SCSI IDs with multiple LUNs.

`/usr/openv/volmgr/bin/sg.build st.conf`
: `-mt` *max_target*
`-ml` *max_lun*
`-st` *st.conf_file*
Updates `st.conf` and generates wide SCSI IDs with multiple LUNs.

`/usr/openv/volmgr/bin/sg.build sg.conf`
: `-mt` *max_target*
`-ml` *max_lun*
`-sc` *sg.conf_file*
Updates `sg.conf` and generates wide SCSI IDs with multiple LUNs.

`/usr/openv/volmgr/bin/sg.build sg.links`
: `-mt` *max_target*
`-ml` *max_lun*
`-sl` *sg.links_file*
Updates `sg.links` and generates wide SCSI IDs with multiple LUNs.

`/usr/openv/volmgr/bin/sgscan` *all*

Scans all connected devices with a SCSI inquiry and provides correlation between physical and logical devices using all device files in `/dev/sg`.

`/usr/openv/volmgr/bin/scsi_command -d /dev/sg/`*sg_id* `-disk`

Changes the device type (stored in the drive's nonvolatile memory) from optical memory to disk.

Where *sg_id* is the logical identifier assigned to the optical disk drive for use by the sg driver. See "Configuring SCSI Robotic Controls" on page 33 for information on determining the logical identifier.

`boot -r`

Reboot the system with the reconfigure option (`-r`) to allow a drive to be recognized as a disk drive during system initialization by the kernel's SCSI disk (sd) driver.

`/usr/openv/volmgr/bin/vmconf`

Provided with Media Manager, this script eases device setup in less complex configurations.

IBM RS6000 Running AIX 4.1.5/4.2/4.3/4.3.1/4.3.2

This chapter describes how to configure devices for use with Media Manager on an IBM RS6000 system. You configure drives and robots using one of the available Media Manager administrative interfaces.

The topics covered are as follows:

- RS6000 AIX Adapter Number Conventions
- Installing the SCSI Pass-Through Driver
- Configuring Robotic Controls
- Configuring Tape Drives
- Configuring Optical Disk Drives
- Command Summary

To determine which drive and robot types are supported, refer to the *NetBackup Release Notes - UNIX*.

Observe the following points when performing the configuration:

- Attach all peripherals and reboot the system before configuring devices. Many of these steps may be accomplished using `smit` (the System Management Interface Tool).
- To obtain error and debugging information about devices and robotic software daemons, the `syslogd` daemon must be configured to be in effect. See `syslogd(1)` for more information.

- VERITAS does not recommend or support the use of single-ended to differential SCSI converters on Media Manager controlled devices. You may encounter problems if you use these converters.
- You can copy code examples from the on-line version of this manual, located in `/usr/openv/volmgr/dev_config_guide.txt`. This file is installed with Media Manager software.

RS6000 AIX Adapter Number Conventions

The location code for an adapter consists of two pairs of digits with the format *AA-BB*; where *AA* identifies the location code of the drawer containing the adapter card and *BB* identifies both the I/O bus and slot containing the card.

A value of 00 for *AA* means that the adapter card is located in the CPU drawer or system unit, depending on the type of system. Any other value for *AA* indicates that the card is located in an I/O expansion drawer; in which case, the value identifies the I/O bus and slot number in the CPU drawer that contains the asynchronous expansion adapter. The first digit identifies the I/O bus with 0 corresponding to the standard I/O bus and 1 corresponding to the optional I/O bus. The second digit identifies the slot on the indicated I/O bus.

The first digit of *BB* identifies the I/O bus containing the adapter card. If the card is in the CPU drawer or system unit, this digit will be 0 for the standard I/O bus or 1 for the optional I/O bus. If the card is in an I/O expansion drawer, this digit is 0. The second digit identifies the slot number on the indicated I/O bus (or slot number in the I/O expansion drawer) that contains the card.

A location code of 00-00 is used to identify the Standard I/O Planar.

Examples

00-05 identifies an adapter card that is in slot 5 of the standard I/O bus in either the CPU drawer or system unit, depending on the type of system.

00-12 identifies an adapter card that is in slot 2 of the optional I/O bus in the CPU drawer.

18-05 identifies an adapter card located in slot 5 of an I/O expansion drawer. The drawer is the one connected to the asynchronous expansion adapter located in slot 8 of the optional I/O bus in the CPU drawer.

Installing the SCSI Pass-Through Driver

Read this topic if you plan to use SCSI-controlled robotic peripherals or Hewlett-Packard 1.2 gigabyte or equivalent model magneto-optical disk drives.

When communicating with SCSI-controlled robotic peripherals on an IBM RS6000 system, Media Manager software utilizes a SCSI pass-through driver called `ovpass`. This driver is also used to set the optical drive type, as documented in"Setting an HP Optical Drive Type in Nonvolatile Memory" on page 75. This pass-through driver is not required if the only peripheral is the IBM 3590 B11 tape stacker.

To install the `ovpass` driver for the first time, enter the following command:

```
/usr/openv/volmgr/bin/driver/install_ovpass
```

To ensure the `ovpass` driver device files are accessible after each system boot, the following command should be placed in the system startup script:

```
/usr/openv/volmgr/bin/driver/mkdev_ovpass
```

Note: The `mkdev_ovpass` command is called by the `rc.veritas.aix` script. You can call this script at system boot time by following the instructions in the Modify Scripts section of the *NetBackup Installation Guide - UNIX*.

To deinstall the `ovpass` driver at a later time, enter the following command:

```
/usr/openv/volmgr/bin/driver/remove_ovpass
```

Note: You cannot use `smit` to configure `ovpass` device files.

Configuring Robotic Controls

Robots are controlled through a SCSI, an RS-232, or a network connection. Configuration for network controlled robotic libraries is discussed in the appendices of the *Media Manager System Administrator's Guide - UNIX*. SCSI and RS-232 control is covered in the following sections.

Configuring SCSI Robotic Controls

Read this topic if you plan to use a robotic storage device that is controlled through a SCSI robotic connection. Supported SCSI robots include the following. See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with the following robot types.

- ODL - Optical Disk Library
- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TL4 - Tape Library 4MM
- TL8 - Tape Library 8MM
- TLD - Tape Library DLT
- TS8 - Tape Stacker 8MM
- TSD - Tape Stacker DLT
- TSH - Tape Stacker Half-inch

Perform the following steps to check for and create the necessary device files.

1. Install the SCSI pass-through driver as explained in "Installing the SCSI Pass-Through Driver" on page 56".
2. Display which SCSI controllers are physically available on your machine by using the following command:

   ```
   /usr/sbin/lsdev -C -c adapter | grep SCSI
   ```

   In the following sample output, SCSI controller 1 (01) has been assigned the logical identifier scsi0:

   ```
   scsi0  Available 00-01    SCSI I/O Controller
   ```

3. Display the SCSI device files that have already been created by using the following command:

   ```
   /usr/sbin/lsdev -C -s scsi
   ```

   The example output shows that two disk drives and one tape drive are configured as follows:

   - hdisk0 is a disk drive at controller 1 (01) and SCSI id 0 (0,0)
   - hdisk1 is a disk drive at controller 1 (01) and SCSI id 1 (1,0)
   - rmt0 is a tape drive at controller 1 (01) and SCSI id 3 (3,0)

   ```
   hdisk0  Available 00-01-00-0,0 400 MB SCSI Disk Drive
   hdisk1  Available 00-01-00-1,0 400 MB SCSI Disk Drive
   rmt0    Available 00-01-00-3,0 Other SCSI Tape Drive
   ```

   If the device files for the SCSI robotic control already exist, they show up in the `lsdev` output as ovpass0, ovpass1, etc. The output for this example does not show any ovpass files so you would have to create them as explained in the next step.

4. If the device files for the desired robotic control SCSI id do not exist, create them with the following command:

   `mkdev -c media_changer -s scsi -t ovpass -p` *ctlr* `-w` *id*`,`*lun*

   Where:

   ctlr is the logical identifier of the drive's SCSI adaptor, such as scsi0, scsi1 or vscsi1.

   id is the SCSI id of the robotic connection.

   lun is the logical unit number of the robotic connection.

   You can display the newly created logical identifier for the device by using the following command:

   ```
   /usr/sbin/lsdev -C -s scsi
   ```

   In this example output, ovpass0 is a SCSI robotic control device file.

   ```
   hdisk0  Available 00-01-00-0,0 400 MB SCSI Disk Drive
   hdisk1  Available 00-01-00-1,0 400 MB SCSI Disk Drive
   rmt0    Available 00-01-00-3,0 Other SCSI Tape Drive
   ovpass0 Available 00-01-5,0    VERITAS Media Changer
   ```

   The path name for these types of device files has the following form:

/dev/*ovpass_id*

Where *ovpass_id* is the logical identifier assigned to the device.

In this example, you use the following device file path:

```
/dev/ovpass0
```

Examples of SCSI Robotic Control Device Files

Example 1

Assume this robot is not a TSD or an HP C1560B. The `ovpass` driver has been installed and the desired SCSI robotic controller is controller 1 at SCSI ID 5, but the device files do not exist.

1. Determine the logical identifier for the SCSI controller as follows:

```
/usr/sbin/lsdev -C -c adapter | grep SCSI
```

The output shows that scsi0 is the logical name for SCSI controller 1.

```
scsi0  Available 00-01    SCSI I/O Controller
```

2. Check if the device files exist for `ovpass` at SCSI ID 5.

```
/usr/sbin/lsdev -C -s scsi
```

```
hdisk0  Available 00-01-00-0,0 400 MB SCSI Disk Drive
rmt0    Available 00-01-00-3,0 Other SCSI Tape Drive
```

The above output shows that the device files exist for tape and disk, but not for the SCSI robotic control at controller 1 (scsi0) and SCSI ID 5 (5,0).

3. Create the device files by using the following command:

```
mkdev -c media_changer -t ovpass -s scsi -p scsi0 -w 5,0
```

4. Display the device files by issuing the `lsdev` command:

```
/usr/sbin/lsdev -C -s scsi
```

```
hdisk0  Available 00-01-00-0,0 400 MB SCSI Disk Drive
hdisk1  Available 00-01-00-1,0 400 MB SCSI Disk Drive
rmt0    Available 00-01-00-3,0 Other SCSI Tape Drive
ovpass0 Available 00-01-5,0    VERITAS Media Changer
```

For this example, use the following device file path to configure the SCSI robot control connected to controller 1 and SCSI ID 5:

```
/dev/ovpass0
```

Example 2

Assume the robot is a DLT2700/DLT4700 (TSD) or an HP C1560B (TL4). The `ovpass` driver has been installed, but the device files for SCSI robotic control at controller 1 with SCSI ID 3 and logical unit number 1, do not exist.

1. Determine the logical identifier for the SCSI controller:

```
/usr/sbin/lsdev -C -c adapter | grep -i SCSI
```

The following output shows that scsi0 is the logical name for SCSI controller 1.

```
scsi0  Available 00-01    SCSI I/O Controller
```

2. Check if the device files exist for `ovpass` at SCSI ID 5.

```
/usr/sbin/lsdev -C -s scsi
```

The following output shows that the device files exist for tape and disk, but not for the SCSI robotic control at controller 1 (scsi0), SCSI ID 3 and logical unit number 1 (3,1):

```
hdisk0  Available 00-01-00-0,0 400 MB SCSI Disk Drive
rmt0    Available 00-01-00-3,0 Other SCSI Tape Drive
```

3. The device files can now be created using the following command:

```
mkdev -c media_changer -t ovpass -s scsi -p scsi0 -w 3,1
```

4. Display the device files by issuing the `lsdev` command:

```
/usr/sbin/lsdev -C -s scsi
```

```
hdisk0  Available 00-01-00-0,0 400 MB SCSI Disk Drive
hdisk1  Available 00-01-00-1,0 400 MB SCSI Disk Drive
rmt0    Available 00-01-00-3,0 Other SCSI Tape Drive
ovpass0 Available 00-01-3,1 VERITAS Media Changer
```

For this example, the device file to use for the TSD SCSI robotic control connected at controller 1 with SCSI ID 3 and logical unit number 1 would be:

```
/dev/ovpass0
```

Example 3

Assume the robot is an STK 9710 connected to a F/W Differential SCSI board and the pass-through driver has been installed. Assume the drives are at SCSI ID's 4 and 5, and the robotics is at SCSI ID 6.

1. Determine the correct scsi controller.

```
lsdev -C | grep scsi

scsi0     Available 00-02       SCSI I/O Controller
ascsi0    Available 00-04       Wide SCSI I/O Controller Adapter
vscsi0    Available 00-04-0,0   SCSI I/O Controller Protocol Device
vscsi1    Available 00-04-0,1   SCSI I/O Controller Protocol Device

lsdev -C -c tape

.
.
rmt2  Available 00-04-01-4,0 Other SCSI Tape Drive
rmt3  Available 00-04-01-5,0 Other SCSI Tape Drive
.
.
```

2. The drives are on Adapter 00-04-01. Therefore, `vscsi1` is the correct adapter for making the ovpass device file as follows:

```
mkdev -c media_changer -t ovpass -s scsi -p vscsi1 -w 6,0
```

Note: Never use the ascsi adapter name.

Example 4 (IBM 3570 B-series Stackers**)**

If there is one drive in the stacker, the robotic control is LUN 1 of the drive's SCSI ID. If there are two drives in the stacker, the robotic control is LUN 1 of the Drive 1 SCSI ID. The SCSI IDs can be set or viewed using the front panel on the stacker. The robotic control for the IBM 3570 B01/B02 is TLD, so if there are two drives, they may be connected to different host systems. If this is the case, the host system which is connected to Drive 1 must also have the robotic control. Also, the library should be in RANDOM mode and BASE configuration. See the Operator's Guide supplied with the unit for information on setting library mode and configuration.

Assume a configuration as follows:

```
lsdev -C -c tape

rmt0 Available 00-02-01-5,0 Other SCSI Tape Drive
rmt0 Available 00-02-01-6,0 Other SCSI Tape Drive
```

If drive 1 is SCSI ID 5, the robotic control for the stacker will be LUN 1 of this SCSI ID. Assuming vscsi1 is the correct adapter, make the passthru device (ovpass) as follows:

```
mkdev -c media_changer -t ovpass -s scsi -p vscsi1 -w 5,1
```

Configuring IBM 3590 Stacker Robotic Controls

Read this topic if you plan to use a TSH - Tape Stacker Half-inch robotic storage device. See the *NetBackup Release Notes - UNIX* for the vendor model associated with the TSH robot type.

Perform the following steps to check for and create the necessary device files:

1. Display the SCSI tape devices configured in the system using the following command:

```
/usr/sbin/lsdev -C -c tape

rmt0  Defined   00-02-00-4,0 Other SCSI Tape Drive
rmt1  Available 00-08-00-6,0 2.3 GB 8mm Tape Drive
.
.
rmt12 Available 00-04-01-6,0 IBM 3590 Tape Drive and Medium
  Changer
```

2. The SCSI robotic path for the IBM 3590 is the same as the no rewind on close tape path. So, when configuring the TSH SCSI robotic path, the robotic control path for the above 3590 would be `/dev/rmt12.1`. The tape drive path would also be `/dev/rmt12.1`.

Configuring RS-232 Robotic Controls

Read this topic if you plan to use a robotic device that is controlled through an RS-232 connection. Supported RS-232 robots include the following:

- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TLD - Tape Library DLT

See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with the above robot types.

Configuring RS-232 controlled devices normally consists of choosing between the two serial console ports and configuring them with the appropriate attributes.

1. Display which RS-232 devices are physically available on your machine by using the following command:

```
/usr/sbin/lsdev -C -c adapter | grep Serial
```

In the following example, serial port 1 (S1) has been given the logical name sa0, while serial port 2 (S2) has been given the name sa1.

```
sa0    Available 00-00-S1 Standard I/O Serial Port 1
sa1    Available 00-00-S2 Standard I/O Serial Port 2
```

2. Display any RS-232 device files that are already configured by using the following command (this example shows one configured device):

   ```
   /usr/sbin/lsdev -C -s rs232
   ```

   The following ouput shows that the device tty0 is configured for serial port 2, as indicated by the S2:

   ```
   tty0 Available  00-00-S2-00 Asynchronous Terminal
   ```

3. If the device files for the desired serial port do not exist, create them with the following command:

   ```
   mkdev -c tty  -t tty -s rs232 -p name -w port -a
   ttyprog_action=off
   ```

   Where:

   name is the logical name for the port, such as sa0 or sa1.

   port reflects the physical port number, such as s1 or s2.

Example of RS-232 Robotic Control Device Files

Assume that the desired serial port is 2 but the device files do not exist.

1. Determine the logical name for serial port 2 as follows:

```
/usr/sbin/lsdev -C -c adapter | grep Serial
```

The following output shows that sa1 is the logical name for serial port 2.

```
sa0    Available 00-00-S1 Standard I/O Serial Port 1
sa1    Available 00-00-S2 Standard I/O Serial Port 2
```

2. Check if the device files exist for serial port 2:

```
/usr/sbin/lsdev -C -s rs232
```

The following output shows that the device files exist for serial port 1 (S1), but not for serial port 2 (S2):

```
tty0 Available  00-00-S1-00 Asynchronous Terminal
```

3. Create the device files with the following command:

```
mkdev -c tty -t tty -s rs232 -p sa0 -w s1 -a
ttyprog_action=off
```

4. Display the device files with the `lsdev` command:

```
/usr/sbin/lsdev -C -s rs232
```

```
tty0 Available  00-00-S1-00 Asynchronous Terminal
tty1 Available  00-00-S2-00 Asynchronous Terminal
```

For this example, enter the following device file path to configure the RS-232 robotic control connected to serial port 2:

```
/dev/tty1
```

Configuring Tape Drives

Read the topics in this section if you plan to use tape drives in your configuration.

Configuring Non-QIC Tape Drives

CAUTION! If you do not configure non-QIC tape drives as variable-length-block devices, NetBackup is able to write data, but may not be able to read it.

As shown by the examples in this section, you must configure non-QIC tape drives as variable-length-block devices if they will be used by Media Manager. Otherwise, NetBackup is able to write data but may not be able to read it correctly. During a read, you see a “not in tar format” error.

The terms *variable length block* or *fixed length block* refers to the behavior of reads and writes and the way the kernel packs physical tape records into logical tape records for an application. Variable-mode devices allow more flexibility in reading previously written tapes. Many tape devices can be accessed in either mode. NetBackup assumes variable length for non-QIC drives. For more information, see `chdev(1)`, `smit(1)` and the System Management Guide. The `smit` application is the most convenient way to change from fixed to variable-length-block devices.

Ensure that the device being used is configured for variable-mode by using the `chdev` command as follows:

```
/usr/sbin/chdev  -l dev -a block_size=0
```

Where *dev* is the logical identifier for the drive (for example: rmt0 or rmt1).

Using Extended-File Marks for Drives

You must configure tape drives to use extended file marks, if those tape drives are capable of supporting them (for example, 8-mm drives). See `chdev(1)` and `smit(1)` for additional information. Otherwise, NetBackup may not be able to use those drives.

Ensure that the device being used is configured for extended file marks as required by Media Manager by using the `chdev` command as follows:

```
/usr/sbin/chdev  -l dev -a extfm=yes
```

Where *dev* is the logical identifier for the drive (for example: rmt0 or rmt1)

Fast-Tape Positioning (locate-block)

For DLT, Exabyte, and half-inch cartridge tape drives, Media Manager supports the SCSI `locate-block` command for positioning tape to a specific block. This improves tape-positioning times over what can be obtained with the alternative, which is the forward-space-record method.

Media Manager uses the `locate-block` command by default unless you disable it by executing:

```
touch /usr/openv/volmgr/database/NO_LOCATEBLOCK
```

With locate-block positioning disabled, NetBackup uses the forward-space-file/record method.

Creating Device Files

When adding tape drives to a Media Manager configuration, you need only specify a no rewind on close device path. These SCSI device files are in the `/dev` directory and have the following format:

`/dev/rmt`*id*.1

Where *id* is the logical identifier assigned to the device by the system.

Perform the following steps to check for and create the necessary device files.

1. Display which SCSI controllers are physically available by using the `lsdev` command as follows:

   ```
   /usr/sbin/lsdev -C -c adapter | grep SCSI
   ```

   This sample output shows that SCSI controller 1 (00-01) has been assigned the logical identifier scsi0.

   ```
   scsi0  Available 00-01    SCSI I/O Controller
   ```

2. Display the SCSI device files that have already been created by using the `lsdev` command.

   ```
   /usr/sbin/lsdev -C -s scsi
   ```

```
hdisk0  Available 00-01-00-0,0 400 MB SCSI Disk Drive
hdisk1  Available 00-01-00-1,0 400 MB SCSI Disk Drive
rmt0    Available 00-01-00-3,0 Other SCSI Tape Drive
```

This example output shows that two disk drives and one tape drive exist as follows:

- hdisk0 is a disk drive at controller 1 (00-01) and SCSI id 0 (0,0)
- hdisk1 is a disk drive at controller 1 (00-01) and SCSI id 1 (1,0)
- rmt0 is a tape drive at controller 1 (00-01) and SCSI id 3 (3,0)

If the device files for the SCSI tape drives exist, they show up in the `lsdev` output as rmt0, rmt1, and so on. The above example output shows rmt0.

For rmt0 and rmt1, you would use the following no rewind on close device files:

```
/dev/rmt0.1
/dev/rmt1.1
```

3. If the device files for the desired tape drive's SCSI ID do not exist, create them using the following `mkdev` command:

```
/usr/sbin/mkdev -c tape -s scsi -t ost -p contr -w id,lun
```

Where:

contr is the logical identifier of the SCSI adapter for the device, such as scsi0 or scsi1.

id is the SCSI ID of the drive connection.

lun is the logical unit number of the drive connection.

An example for an 8-mm drive connected to controller 0 and SCSI ID 5 follows:

```
mkdev -c tape -s scsi -t ost -p scsi0 -w 5,0
```

You can display the newly created logical identifier for the device by using the `lsdev` command.

```
/usr/sbin/lsdev -C -s scsi
```

```
hdisk0  Available 00-01-00-0,0 400 MB SCSI Disk Drive
hdisk1  Available 00-01-00-1,0 400 MB SCSI Disk Drive
rmt0    Available 00-01-00-3,0 Other SCSI Tape Drive
rmt1    Available 00-01-00-5,0 Other SCSI Tape Drive
ovpass0 Available 00-01-6,0    VERITAS Media Changer
```

The rmt1 device file has been created.

4. Ensure that the device being used is configured for variable-mode and extended file marks as required by Media Manager by using the chdev command as follows:

```
/usr/sbin/chdev   -l dev -a block_size=0
/usr/sbin/chdev   -l dev -a extfm=yes
```

Where *dev* is the logical identifier for the drive (for example: rmt0 or rmt1)

No Rewind Device File Example

Assume the device files for the desired SCSI 8-mm tape drive (controller 1, SCSI ID 5) do not exist.

1. Determine the logical identifier for the SCSI controller as follows:

```
/usr/sbin/lsdev -C -c adapter | grep SCSI
```

The following output shows that scsi0 is the logical name for SCSI controller 1.

```
scsi0   Available 00-01     SCSI I/O Controller
```

2. Check if the device files exist for any device at SCSI ID 5.

```
/usr/sbin/lsdev -C -s scsi
```

The following output shows that some device files exist for tape and disk, but not for the 8-mm tape drive at controller 1 (scsi0) and SCSI ID 5 (5,0):

```
hdisk0   Available 00-01-00-0,0 400 MB SCSI Disk Drive
hdisk1   Available 00-01-00-1,0 400 MB SCSI Disk Drive
rmt0      Available 00-01-00-3,0 Other SCSI Tape Drive
```

3. Create the desired device files by using the following mkdev command:

```
mkdev -c tape -t ost -s scsi -p scsi0 -w 5,0
```

4. Display the device files by issuing the following lsdev command:

```
/usr/sbin/lsdev -C -s scsi
```

```
hdisk0   Available 00-01-00-0,0 400 MB SCSI Disk Drive
hdisk1   Available 00-01-00-1,0 400 MB SCSI Disk Drive
rmt0      Available 00-01-00-3,0 Other SCSI Tape Drive
rmt1      Available 00-01-00-5,0 Other SCSI Tape Drive
```

5. To ensure that the tape device is configured for variable-mode and extended file marks, use the following commands:

```
chdev -l rmt1 -a block_size=0
chdev -l rmt1 -a extfm=yes
```

Enter the following device file path to configure the 8-mm drive connected to controller 1 and SCSI ID 5:

```
/dev/rmt1.1
```

Using Multiple Tape Densities

After creating the necessary device files for your tape drives you may want to make use of nondefault densities on drives that support them (for example, Exabyte 8500C tape drives).

There are two configurable densities available for all tape drives, although not all tape drives support multiple densities. The default density for both density setting 1 and density setting 2 is 0, which means maximum density. To modify either of the density settings, you can use `smit(1)` or commands similar to the following:

```
chdev   -l tapedev -a density_set_1=density
chdev   -l tapedev -a density_set_2=density
```

Where:

tapedev is the logical identifier for the drive, such as rmt0 or rmt1

density is the decimal number representing the desired density

To use density setting 1, use the following no rewind on close device file:

```
/dev/rmt*.1
```

To use density setting 2, use the following no rewind on close device file:

```
/dev/rmt*.5
```

Adding HP 4-mm Drives and HP C1560B DAT Autoloaders

To support HP (Hewlett-Packard) 4-mm DAT tape drives and HP C1560B DAT Autoloaders use the following hardware (tape drive) switch settings. Other combinations may work, but these are the settings that were functional during testing with an HP 35480 tape drive and HP C1560B DAT Autoloader.

On=1, Off=0

Switch	Setting
1	1
2	1
3	1
4	1
5	1
6	1
7	0
8	0

Adding Sony AIT Drives

Read this section if you plan to use Sony AIT tape drives in your configuration.

No Rewind Device Files

When adding tape drives to a Media Manager configuration, you need only specify a no rewind on close device path. To display the no rewind device files that are configured on your system, use the `lsdev` command as follows:

```
/usr/sbin/lsdev -C -s scsi
```

```
rmt6 Available 00-03-01-6,0 Other SCSI Tape Drive
```

Using the `lsdev` output, if the drive is connected to SCSI ID 6 of adapter 3, the device path you use follows:

```
/dev/rmt0.1
```

Dip Switch Settings

Sony AIT drives have 8 dip switches located on the bottom of the drive. It is important to set these switches correctly, even if it means taking the drives out of robots and checking them.

Some robots (for example, SpectraLogic) provide a way to set the drive switches from the robot itself. For SpectraLogic robots, it doesn't matter what the drive switches are. The Treefrog (215) robot has a dial in the back to set the appropriate OS. The Bullfrog (10000) robot has a means of setting the OS through the touchscreen.

Depending on the version of the AIT drive, drives are shipped from Sony with one of the following two settings as shown below:

Note: Robot vendors and hardware resellers may change the default drive switch settings.

On=1 and Off=0.

Switch	Setting
1	0
2	0
3	0
4	0
5	0
6	0
7	1
8	1

Switch	Setting
1	0
2	0
3	0
4	0
5	1
6	0
7	1
8	0

Switches 1 thru 4 are critical for setting the OS type. Usually, switches 5 - 8 can be left set at the default. For AIX, use the following switch settings:

Switch	Setting
1	1
2	0
3	0
4	0

You can use the following command to determine the correct dip switch settings without removing the drives and checking them:

```
/usr/openv/volmgr/bin/scsi_command -d /dev/rmt0.1 -ait
```

The output is as follows:

```
Physical AIT drive switch setting = 0x1 (IBM RS6000 - AIX
- disconnect enabled)

Logical AIT drive switch setting = 0xff (Not set, physical
setting in effect)
```

The above example was an AIT drive in a ADIC Grau library. The drive was taken out and set to the AIX switch settings.

Configuring Optical Disk Drives

When adding optical disk drives to a Media Manager configuration, you need specify only a character device path. Optical disk character device files are located in the `/dev` directory and have the following format:

`/dev/rhdisk`*id*

Where *id* is the logical identifier assigned to the device by the system.

Note: To use Hewlett-Packard optical disk drives, the system must recognize the optical drives as disk drives at system boot time. If you are adding Hewlett-Packard 1.2 gigabyte or equivalent model magneto-optical disk drives to an AIX system, the system may not recognize them as disk drives, and thus cannot use them. See “Setting an HP Optical Drive Type in Nonvolatile Memory” on page 75 for information on correcting this condition.

Creating Device Files

Perform the following steps to check for and create the necessary device files.

1. Display which SCSI controllers are physically available on your machine by using the following `lsdev` command:

   ```
   /usr/sbin/lsdev -C -c adapter | grep SCSI
   ```

   This sample output shows that SCSI controller 1 (00-01) has been assigned the logical identifier scsi0.

   ```
   scsi0   Available 00-01     SCSI I/O Controller
   ```

2. Display the SCSI device files that have already been created by using the following `lsdev` command.

   ```
   /usr/sbin/lsdev -C -s scsi
   ```

   ```
   hdisk0  Available 00-01-00-0,0 400 MB SCSI Disk Drive
   hdisk1  Available 00-01-00-1,0 400 MB SCSI Disk Drive
   rmt0    Available 00-01-00-3,0 Other SCSI Tape Drive
   ```

   The above example output shows that two disk drives and one tape drive exist:

   - hdisk0 is a disk drive at controller 1 (00-01) and SCSI id 0 (0,0)

- hdisk1 is a disk drive at controller 1 (00-01) and SCSI id 1 (1,0)
- rmt0 is a tape drive at controller 1 (00-01) and SCSI id 3 (3,0)

If the device files for the SCSI optical disk drives exist, they show up in the `lsdev` output as hdisk0, hdisk1, and so on.

For hdisk0, you would use the following device path:

```
/dev/rhdisk0
```

3. If the device files for the desired optical drive's SCSI ID do not exist, you can create them with the following command:

   `mkdev -c disk -s scsi -t osdisk -p` *controller* `-w` *id*`,`*lun*

   Where:

   controller is the logical identifier of the device's SCSI adapter, such as scsi0 or scsi1.

   id is the SCSI id of the drive connection.

   lun is the logical unit number of the drive connection.

   An example for an optical disk drive on controller 1 and SCSI ID 5 would be as follows:

   ```
   mkdev -c disk -t osdisk -s scsi -p scsi0 -w 5,0
   ```

4. You can display the newly created logical identifier for the device by using the following command:

   ```
   /usr/sbin/lsdev -C -s scsi
   ```

   ```
   hdisk0  Available 00-01-00-0,0 400 MB SCSI Disk Drive
   hdisk1  Available 00-01-00-1,0 400 MB SCSI Disk Drive
   rmt0    Available 00-01-00-3,0 Other SCSI Tape Drive
   hdisk2  Available 00-01-00-5,0 Other SCSI Disk Drive
   ovpass0 Available 00-01-6,0    VERITAS Media Changer
   ```

   The device files for `hdisk2` have been created and you can now use them.

Examples of Optical Disk Device Files

Assume the device files for the desired optical disk drive (controller 1, SCSI ID 5) do not yet exist.

1. Determine the logical identifier for the SCSI controller as follows:

```
/usr/sbin/lsdev -C -c adapter | grep SCSI
```

The output shows that scsi0 is the logical name for SCSI controller 1.

```
scsi0  Available 00-01    SCSI I/O Controller
```

2. Check to see if the device files exist for `ovpass` at SCSI ID 5.

```
/usr/sbin/lsdev -C -s scsi
```

The output shows that some device files exist for tape and disk, but not for the optical disk drive at controller 1 (scsi0) and SCSI ID 5 (5,0).

```
hdisk0  Available 00-01-00-0,0 400 MB SCSI Disk Drive
hdisk1  Available 00-01-00-1,0 400 MB SCSI Disk Drive
rmt0    Available 00-01-00-3,0 Other SCSI Tape Drive
```

3. Create device files for the optical disk drive on controller 1 at SCSI ID 5 by using the following command:

```
mkdev -c disk -t osdisk -s scsi -p scsi0 -w 5,0
```

4. Display the device files by issuing the `lsdev` command:

```
/usr/sbin/lsdev -C -s scsi
```

```
hdisk0  Available 00-01-00-0,0 400 MB SCSI Disk Drive
hdisk1  Available 00-01-00-1,0 400 MB SCSI Disk Drive
rmt0    Available 00-01-00-3,0 Other SCSI Tape Drive
hdisk2  Available 00-01-00-5,0 Other SCSI Disk Drive
```

5. Enter the following character device file path to configure the optical disk drive connected to controller 1 and SCSI ID 5:

```
/dev/rhdisk2
```

Setting an HP Optical Drive Type in Nonvolatile Memory

To use Hewlett-Packard optical disk drives, the system must recognize the optical drives as disk drives at system boot time. If you are adding Hewlett-Packard 1.2 gigabyte or equivalent model magneto-optical disk drives to an AIX system, the system may not recognize them as disk drives, and thus cannot use them.

To detect whether the system recognizes the optical drives as disk drives, execute the following command after system boot.

```
/usr/sbin/lsdev -C -s scsi
```

```
hdisk0  Available 00-00-0S-0,0 2.2 GB SCSI Disk Drive
rmt0    Available 00-00-0S-3,0 Other SCSI Tape Drive
omd0    Defined   00-00-0S-6,0 Other SCSI Read/Write Optical Drive
ovpass0 Available 00-00-0S-2,0 VERITAS Media Changer
```

If you see the appropriate controller and SCSI ID combination for the optical drive listed as *Other SCSI Disk Drive*, the system recognizes the drive as a disk drive. If not, perform the following steps:

1. Install the ovpass driver if it is not already installed. See "Installing the SCSI Pass-Through Driver" on page 56 for information on how to install the ovpass driver.

2. Create the ovpass device file for the optical drive so that the ovpass driver can be used to communicate with the optical drive.

   a. Display the SCSI device files that have already been created by using the following command:

```
/usr/sbin/lsdev -C -s scsi
```

```
hdisk0  Available 00-00-0S-0,0 2.2 GB SCSI Disk Drive
rmt0    Available 00-00-0S-3,0 Other SCSI Tape Drive
omd0    Defined   00-00-0S-6,0 Other SCSI Read/Write Optical Drive
ovpass0 Available 00-00-0S-2,0 VERITAS Media Changer
```

   The above example shows that a disk drive, a tape drive, an optical drive, and SCSI robotic control are configured:

   - hdisk0 is a disk drive at controller 1 (00) and SCSI id 0 (0,0)
   - rmt0 is a tape drive at controller 1 (00) and SCSI id 3 (3,0)
   - omd0 is an optical drive at controller 1 (00) and SCSI id 6 (6,0)
   - ovpass0 refers to the SCSI robotic control for controller 1 (00) and SCSI id 2 (2,0)

   b. Make the device files for the optical drive by using the following command:

```
mkdev -c media_changer -s scsi -t ovpass -p ctlr -w id,lun
```

   Where:

   ctlr is the logical identifier of the drive's SCSI adapter, such as scsi0 or scsi1.

   id is the SCSI id of the optical drive (not the robotic connection).

lun is the logical unit number of the optical drive.

For example:

```
mkdev -c media_changer -s scsi -t ovpass -p scsi 0 -w 6,0
```

Use the following command to obtain the logical identifier for the optical drive you just created.

```
/usr/sbin/lsdev -C -s scsi
```

c. Verify the temporary `ovpass` device file created in step b.

```
/usr/openv/volmgr/bin/scsi_command -d /dev/ovpass_id
    -inquiry
```

Where *ovpass_id* is the logical identifier assigned to the temporary device. For example, if the temporary `ovpass` device was ovpass2, enter:

```
/usr/openv/volmgr/bin/scsi_command -d /dev/ovpass2
    -inquiry
removable device type c_8h_HP
```

3. Use the following command to change the device type (stored in the drive's nonvolatile memory) from *optical memory* to *disk*. The format of the command is as follows:

```
/usr/openv/volmgr/bin/scsi_command -d /dev/ovpass_id -disk
```

Where *ovpass_id* is the logical identifier assigned to the device

For example:

```
/usr/openv/volmgr/bin/scsi_command -d /dev/ovpass1 -disk
```

4. Remove the `ovpass` device files and the optical drive that were created by using `rmdev` command as in the following:

```
rmdev -l ovpass_id -d
rmdev -l optical_drive_id -d
```

Where:

ovpass_id is the logical identifier assigned to the device.

optical_drive_id is the optical drive identifier assigned to the optical drive.

For example:

```
rmdev -l ovpass1 -d
rmdev -l omd0 -d
```

5. Reboot the system to allow the drive to be recognized as a disk drive by the kernel's SCSI disk driver during system initialization.

   The optical drive should be displayed as:

   `hdisk` *logical_number*

   Where *logical_number* is the logical number assigned to the drive by the system.

   For example:

   ```
   /usr/sbin/lsdev -C -s scsi
   ```

   The following listing shows a disk drive, tape drive, robotic control, and optical drive.

   ```
   hdisk0  Available 00-00-0S-0,0 2.2 GB SCSI Disk Drive
   rmt0    Available 00-00-0S-3,0 Other SCSI Tape Drive
   ovpass0 Available 00-00-0S-2,0 VERITAS Media Changer
   hdisk1  Available 00-00-0S-6,0 Other SCSI Disk Drive
   ```

Command Summary

The following is a summary of commands that may be useful when configuring devices. See the procedures in this chapter for examples of their usage.

`/usr/openv/volmgr/bin/driver/install_ovpass`
: Installs the ovpass driver for the first time.

`/usr/openv/volmgr/bin/driver/remove_ovpass`
: Uninstalls the ovpass driver.

`/usr/openv/volmgr/bin/driver/mkdev_ovpass`
: Place this command in the system startup script to ensure that the `ovpass` driver device files are accessible after each system boot.

`/usr/sbin/lsdev -C -c adapter | grep` *type*
: Displays adapters that are physically available on your machine. *type* defines the type of adapter displayed: `SCSI` displays SCSI adapters and `Serial` displays RS-232 adapters.

`/usr/sbin/lsdev -C -s` *filetype*
: Displays the device files that have been created, where `scsi` displays SCSI and `rs232` displays RS-232.

`mkdev -c media_changer -s scsi -t ovpass -p` *controller* `-w` *id*`,`*lun*
: Creates device files for the robotic control SCSI ID.
Where *controller* is the logical identifier of the drive SCSI adaptor (such as scsi0 or scsi1), *id* is the SCSI ID of the robotic connection, and *lun* is the logical unit number of the robotic connection.

`mkdev -c disk -s scsi -t osdisk -p` *controller* `-w` *id*`,`*lun*
: Creates device files for optical disk drives.
Where *controller* is the logical identifier of the drive SCSI adaptor (such as scsi0 or scsi1), *id* is the SCSI ID of the robotic connection, and *lun* is the logical unit number of the robotic connection.

`mkdev -c tape -s scsi -t ost -p` *controller* `-w` *id*`,`*lun*
: Creates device files for tapes.
Where *controller* is the logical identifier of the drive SCSI adaptor (such as scsi0 or scsi1), *id* is the SCSI ID of the robotic connection, and *lun* is the logical unit number of the robotic connection.

`mkdev -c tty -t tty -s rs232 -p` *ID* `-w` *port* `-a ttyprog_action=off`
Creates device files for the desired serial port.
Where *ID* is the logical identifier for the port, such as sa0 or sa1 and *port* reflects the physical port number, such as s1 or s2.

`/usr/sbin/chdev -l` *dev* `-a block_size=0`
Configures the drive with logical identifier specified by *dev* (for example: rmt0) to variable mode.

`/usr/sbin/chdev -l` *dev* `-a extfm=yes`
Configures the drive with logical identifier specified by *dev* (for example: rmt0) for extended file marks.

`/usr/openv/volmgr/bin/scsi_command -d /dev/`*ovpass_id* `-disk`
Used for HP optical disk drives to change the device type (stored in the drive's nonvolatile memory) from optical memory to disk.
Where *ovpass_id* is the logical identifier assigned to the device.

`/usr/openv/volmgr/bin/vmconf`
Provided with Media Manager, this script eases device setup in less complex configurations.

`/etc/lsattr -l` *dev* `-E -H`
Displays device information, where *device* is the name of the device (for example, rmt1).

HP9000-700 Running HP-UX 10.20/11.0

This chapter shows how to configure devices for use with Media Manager on an HP9000-700 system. You configure drives and robots using one of the available Media Manager administrative interfaces.

The topics included are as follows:

- Configuring Robotic Controls
- Configuring Tape Drives
- Configuring Optical Disk Drives
- Command Summary

To determine which drive and robot types are supported, refer to the *NetBackup Release Notes - UNIX*.

Be sure to observe the following items when performing the configuration described in this chapter:

- VERITAS does not recommend or support the use of single-ended to differential SCSI converters on Media Manager controlled devices. You may encounter problems if you use these converters.
- You can copy code examples from the on-line version of this manual, located in `/usr/openv/volmgr/dev_config_guide.txt`. This file is installed with Media Manager software.

Configuring Robotic Controls

Robots are controlled through a SCSI, an RS-232, or a network connection. Configuration for network controlled robotic libraries is discussed in the appendixes of the *Media Manager System Administrator's Guide - UNIX*. SCSI and RS-232 control is covered in the following sections.

Configuring SCSI Robotic Controls

Read this topic if you plan to use a robotic storage device that is controlled through a SCSI robotic connection. Supported SCSI robots include.

- ODL - Optical Disk Library
- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TL4 - Tape Library 4MM
- TL8 - Tape Library 8MM
- TLD - Tape Library DLT
- TS8 - Tape Stacker 8MM
- TSD - Tape Stacker DLT

When communicating with SCSI-controlled robotic peripherals, Media Manager robotic software utilizes the generic (user mode) SCSI pass-through driver. You do not have to reconfigure the HP-UX kernel to use this driver on HP9000-700 systems, since the generic SCSI driver is part of basic HP-UX.

If the devices do not exist, you can create device files by using the `mknod` command (see the `scsi_ctl(7)` man page) as follows:

```
mkdir /dev/sctl
cd /dev/sctl
/etc/mknod ccontrollertargetdlun c 203 0xiitl00
```

Where:

controller is the Instance number of the controlling bus. The Instance value is displayed in `ioscan -f` output under column I of the controller entry (ext_bus in the Class column).

target is the SCSI ID of the robotic control.

lun is the SCSI logical unit number and should be 0 for all robots, except DLT2700, DLT4700, HP C1560B, and a few other robots where *lun* must be 1.

ii are two hexadecimal digits that identify the controlling bus interface card by its Instance number (same as controller).

t is one hexadecimal digit representing the SCSI ID.

l is one hexadecimal digit representing the SCSI LUN.

Examples of SCSI Robotic Control Device Files

Example 1

If the robotic control for an Exabyte 10i (TS8) is connected to a SCSI controller with Instance number 0 at SCSI ID 5, LUN 0 and the `/dev/sctl` files exist, the device file path to use is:

```
/dev/sctl/c0t5d0
```

If the `/dev/sctl` files do not exist, the commands to create the device file are:

```
cd /dev/sctl
/etc/mknod c0t5d0 c 203 0x005000
```

This creates the following device file, which you specify to Media Manager:

```
/dev/sctl/c0t5d0
```

Example 2

If the robotic control for an HP Optical Disk Library (ODL) is on an EISA adapter with Instance number 2 at SCSI ID 3, LUN 0, the commands to create the device file are:

```
cd /dev/sctl
/etc/mknod c2t3d0 c 203 0x023000
```

This creates the following device file, which you specify to Media Manager:

```
/dev/sctl/c2t3d0
```

Example 3

If the robotic control for a DLT2700 or DLT4700 is connected to the controller with Instance number 0 at SCSI ID 3, LUN 1, the commands to create the device file are as follows:

```
cd /dev/sctl
/etc/mknod c0t3d1 c 203 0x003100
```

This creates the following device file, which you specify to Media Manager:

```
/dev/sctl/c0t3d1
```

Configuring RS-232 Robotic Controls

Read this topic if you plan to use a robotic storage device that is controlled through an RS-232 connection. Supported RS-232 robots include:

- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TLD - Tape Library DLT

Configuring RS-232 controlled devices consists of determining the Instance number of the controlling interface and the port that will be used.

Device names for RS-232 connections use the following format:

`/dev/tty`*`I`*`p`*`Port`*

Where:

I is the Instance number of the controlling interface.

Port is the port number.

You can determine the Instance number by using the following command:

```
ioscan -f -C tty
```

In the following output the interface Instance number, found under column I, is 0.

```
Class I  H/W Path Driver S/W State H/W Type    Description
=============================================================

tty   0  2/0/4    asio0  CLAIMED   INTERFACE    Built-in RS-232C
```

Example of RS-232 Robotic Control Device File

If the RS-232 robotic control is connected to port 0 of the built-in RS-232C interface, and the Instance number of the interface from `ioscan` is 0, the device file path you use follows:

```
/dev/tty0p0
```

Configuring Tape Drives

Using Berkeley Style Close

The examples in this section show *Berkeley-style close* for tape drives as indicated by the letter `b` after the density specification. It is mandatory to specify Berkeley-style close for tape devices that you configure under Media Manager.

The terms *Berkeley-style close* and *AT&T style close* refer to where a tape is left logically positioned after a close operation (in relation to a tape mark). One style leaves an application logically positioned before a tape mark and the other leaves it after. Applications must assume where the tape is left after a close in order to establish the correct orientation the next time they do a tape-position or read operation. Some operating systems allow tape devices to be configured with either type of close. NetBackup assumes it is using Berkeley-style close on an HP9000-700.

Device Files

When adding tape drives to a Media Manager configuration, you need specify only a no rewind on close device path. To determine if the tape device files exist on your system, check the `/dev/rmt` directory. No rewind on close device files have the following format:

`/dev/rmt/c`*Controller*`t`*Target*`d`*Unit*_Density_`nb`

Where:

Controller is the Instance number of the controlling bus. The Instance value is displayed in `ioscan -f` output under column I of the controller entry (ext_bus in the Class column).

Target is the SCSI ID of the tape drive.

Unit is the SCSI logical unit number (LUN) of the drive. This is usually 0.

Density is the desired density of the tape drive.

For example:

- `D8MM_8500C` for 8505 drives
- `DDS2C` for DDS2 compression drives

See the `mt(7)` man page for an explanation of the tape drive densities.

If the desired tape device files do not exist, you can create them using `sam`, the system administration manager, or the `mksf(1M)` command. The following is an example using `mksf`:

`mksf -C tape -H` *H/W Path* `-b` *density* `-u -n`

Where:

H/W Path is the H/W Path of the tape drive as specified by `ioscan`.

density is the density string to use for the drive as defined in the `mksf(1M)` and `mt(7)` man pages.

Examples of No Rewind Device Files

Example 1

Assume that the desired Exabyte 8505 tape drive is on the built-in SCSI interface at SCSI ID 4 and the `ioscan` command shows the following output:

```
ioscan -f
```

```
Class    I H/W Path   Driver      S/W State H/W Type  Description
=============================================================

bc       0            root        CLAIMED   BUS_NEXUS
graphics 0  1         graph3      CLAIMED   INTERFACE Graphics
ba       0  2         bus_adapter CLAIMED   BUS_NEXUS Core I/O Adapter
ext_bus  0  2/0/1     c700        CLAIMED   INTERFACE Built-in SCSI
target   2  2/0/1.4   tgt         CLAIMED   DEVICE
tape     5  2/0/1.4.0 stape       CLAIMED   DEVICE    EXABYTE EXB-8505
.
.
.
```

The Instance number for the controlling bus is 0, and the H/W path for the tape drive is 2/0/1.4.0. The command to create the device file follows:

```
mksf -C tape -H 2/0/1.4.0 -b D8MM_8500C -u -n
```

This creates the following device file, which you specify to Media Manager:

```
/dev/rmt/c0t4d0D8MM8500Cnb
```

You can display the device files for the drive with the following command:

```
ioscan -f -H 2/0/1.4.0 -n
```

```
Class I  H/W Path   Driver  S/W State  H/W Type  Description
===================================================================

tape  5  2/0/1.4.0 stape   CLAIMED   DEVICE   EXABYTE EXB-85058SQANXR1
                    /dev/rmt/3m         /dev/rmt/c0t4d0BESTb
                    /dev/rmt/3mb        /dev/rmt/c0t4d0BESTn
                    /dev/rmt/3mn        /dev/rmt/c0t4d0BESTnb
                    /dev/rmt/3mnb       /dev/rmt/c0t4d0D8MM8500Cb
                    /dev/rmt/c0t4d0BEST /dev/rmt/c0t4d0D8MM8500Cnb
```

Example 2

Assume that the desired DAT (4mm) tape drive with compression is on an EISA adapter at SCSI 3 and ioscan shows the following:

```
ioscan -f
```

```
Class     I  H/W Path  Driver      S/W State H/W Type     Description
===================================================================
bc        0            root        CLAIMED   BUS_NEXUS
graphics  0  0         graph3      CLAIMED   INTERFACE    Graphics
ba        0  2         bus_adapter CLAIMED   BUS_NEXUS    Core I/O Adapter
ext_bus   0  2/0/1     c700        CLAIMED   INTERFACE    Built-in SCSI
.
.
.
ba        1  4         eisa        CLAIMED   BUS_NEXUS    EISA Adapter
ext_bus   2  4/0/1      c700        CLAIMED   INTERFACE   EISA card HWP0C80
target    9  4/0/1.3   tgt         CLAIMED   DEVICE
tape      5  4/0/1.3.0 stape       CLAIMED   DEVICE       HP       C1533A
```

The Instance number for the controlling bus (ext_bus) is 2 and the H/W path for the tape drive is 4/0/1.3.0. The command to create the device file for this tape drive is:

```
mksf -C tape -H 4/0/1.3.0 -b DDS2C -u -n
```

This creates the following device file, which you specify to Media Manager:

```
/dev/rmt/c2t3d0DDS2Cnb
```

Switch Settings for HP C1533A 4-mm DAT Drives

If you have standalone or robotic 4-mm drives that are model HP C1533A, you may have to change the switch settings on the bottom of the drive. This is the drive that comes in the HP C1560B (48AL) DAT Autoloader

If the C1533A drive or HP C1560B autoloader was purchased from Hewlett Packard, the default switch settings should work. These default settings, as documented by Hewlett Packard, are as follows:

On=1, Off=0

Switch	Setting
1	1
2	1
3	0
4	1
5	1
6	1
7	1
8	1

However, if the drive or autoloader were purchased from another vendor, and that vendor changed the switch settings, you will have to set the switches as shown above.

You may also have to make this change to HP C1533A drives in non-Hewlett Packard 4-mm robots.

Configuring Optical Disk Drives

When adding optical disk drives to a Media Manager configuration, you only need to specify a character device path. Optical disk character device files are found in the `/dev/rdsk` directory and have the following format:

`/dev/rdsk/c`*Controller*`t`*Target*`d`*Unit*

Where:

Controller is the Instance number of the controlling bus. The Instance value is displayed in `ioscan -f` output under the column I of the controllers entry (ext_bus in the Class column).

Target is the SCSI ID of the drive.

Unit is the SCSI logical unit number (LUN) of the drive and is usually 0.

If the desired character device files do not exist, create them with the `mksf` command. The following is an example:

`mksf -C disk -H` *H/W Path* `-r`

Where *H/W Path* is the H/W Path of the disk drive as specified by `ioscan`.

Examples of Optical Disk Device Files

Example 1

Assume that the desired optical disk drive is on the built-in SCSI interface at SCSI ID 4 and `ioscan` shows the following:

```
ioscan -f

Class     I  H/W Path   Driver  S/W State H/W Type  Description
=============================================================
ext_bus   0  2/0/1      c700    CLAIMED   INTERFACE Built-in SCSI
target    4  2/0/1.4    tgt     CLAIMED   DEVICE
disk      1  2/0/1.4.0  sdisk   CLAIMED   DEVICE    HP      C1716T
.
.
.
```

The Instance number for the controlling bus is 0, and the H/W path for the optical disk drive is 2/0/1.4.0. The command to create the device file for drive follows:

```
mksf -C disk -H 2/0/1.4.0 -r
```

This creates the following device file, which you specify to Media Manager:

```
/dev/rdsk/c0t4d0
```

Example 2

Assume that the desired optical disk drive is on an EISA interface at SCSI ID 3 and `ioscan` shows the following:

```
ioscan -f
```

```
Class     I  H/W Path    Driver       S/W State  H/W Type     Description
=============================================================================
bc        0              root         CLAIMED    BUS_NEXUS
graphics  0  0           graph3       CLAIMED    INTERFACE    Graphics
ba        0  2           bus_adapter  CLAIMED    BUS_NEXUS    Core I/O Adapter
ext_bus   0  2/0/1       c700         CLAIMED    INTERFACE    Built-in SCSI
.
.
.
ba        1  4           eisa         CLAIMED    BUS_NEXUS    EISA Adapter
ext_bus   2  4/0/1       c700         CLAIMED    INTERFACE    EISA card HWP0C80
target    9  4/0/1.3     tgt          CLAIMED    DEVICE
disk      5  4/0/1.3.0   sdisk        CLAIMED    DEVICE       HP C1716T
```

The Instance number for the controlling bus is 2, and the H/W path for the optical disk drive is 4/0/1.3.0. The command to create the device file for drive follows:

```
mksf -C disk -H 4/0/1.3.0 -r
```

This creates the following device file, which you specify to Media Manager:

```
/dev/rdsk/c2t3d0
```

Command Summary

The following is a summary of commands that may be useful when configuring devices. See the procedures in this chapter for examples of their usage.

`ioscan -f`

Displays information about the physical interfaces available in your system. For example, it shows the hardware path and the Instance number for the controlling bus.

`/etc/mknod c`*`controller`*`t`*`target`*`d`*`lun`* `c 203 0x`*`iitl`*`00`

Creates device files for SCSI robotic controlled robotics.
controller is the Instance number of the controlling bus. The Instance value is displayed in `ioscan -f` output under column I of the controller entry (ext_bus in the Class column).
target is the SCSI ID of the robotic control.
lun is the SCSI logical unit number and should be 0 for most robots. Exceptions are Quantum DLT2700 and DLT2700, HP C1560B, and a few other robots where *lun* must be 1.
ii is two hexadecimal digits that identify the controlling bus interface card by its Instance number (same as controller).
t is one hexadecimal digit representing the SCSI ID.
l is one hexadecimal digit representing the SCSI LUN.

`mksf -C tape -H` *H/W Path* `-b` *density* `-u -n`

Creates device files for tape drives.
H/W Path is the hardware path of the tape drive as specified by `ioscan`.
density is the density string to use for the drive as defined in the `mksf(1M)` and `mt(7)` man pages.

`mksf -C disk -H` *H/W Path* `-r`

Creates device files for optical disk drives.
Where *H/W Path* is the H/W Path of the disk drive as specified by `ioscan`.

HP9000-800 Running HP-UX 10.20/11.0

This chapter shows how to configure devices for use with Media Manager on an HP9000-800 system. You configure drives and robots using one of the available Media Manager administrative interfaces.

The major topics included are as follows:

- Configuring Robotic Controls
- Configuring Tape Drives
- Configuring Optical Disk Drives
- Command Summary

To determine which drive and robot types are supported, refer to the *NetBackup Release Notes - UNIX*.

Be sure to observe the following items when performing the configuration described in this chapter:

- VERITAS does not recommend or support the use of single-ended to differential SCSI converters on Media Manager controlled devices. You may encounter problems if you use these converters.
- You can copy code examples from the on-line version of this manual, located in `/usr/openv/volmgr/dev_config_guide.txt`. This file is installed with NetBackup and Media Manager software.

Configuring Robotic Controls

Robots are controlled through a SCSI, an RS-232, or a network connection. Configuration for network controlled robotic libraries is discussed in the appendices of the *Media Manager System Administrator's Guide - UNIX*. SCSI and RS-232 control is covered in the following sections.

Configuring SCSI Robotic Controls

Read this topic if you plan to use a robotic storage device that is controlled through a SCSI robotic connection. Supported SCSI robots include the following. See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with the following robot types:

- ODL - Optical Disk Library
- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TL4 - Tape Library 4MM
- TL8 - Tape Library 8MM
- TLD - Tape Library DLT
- TS8 - Tape Stacker 8MM
- TSD - Tape Stacker DLT

Determining Which Pass-Through Driver to Configure

When communicating with SCSI-controlled robotic peripherals, Media Manager robotic software uses the spt or sctl SCSI pass-through driver. The driver that is used depends on the type of SCSI interface on the system.

The two types of SCSI interface are as follows:

- Interfaces that use the scsi1/scsi3 bus-adapter driver require the spt pass-through driver. The 28655A SCSI interface is in this category.
- Interfaces that use the c700/c720 bus-adapter driver require the sctl pass-through driver. The GSC built-in SCSI interface, and some add-on cards for HP9000-800 D, K, T, and V series systems are in this category. When attaching an autochanger device to a GCS interface and using the sctl

driver, the schgr device driver must also be installed. Without this driver installed, the system will not bind the driver to the device. See the `autochanger(7)` man page.

To determine the type of interface on your system, use the `ioscan -f` command as shown in the examples below.

Example 1: 28655A SCSI Interface (spt driver)

```
ioscan -f
```

```
Class   I  H/W Path Driver S/W State   H/W Type     Description
=============================================================================
bc      0           root    CLAIMED   BUS_NEXUS
bc      1 56        bc      CLAIMED   BUS_NEXUS Bus Converter
ext_bus 0 56/52     scsi1   CLAIMED   INTERFACE HP 28655A - SCSI Interface
target  0 56/52.2   target  CLAIMED   DEVICE
tape    0 56/52.2.0 tape2   CLAIMED   DEVICE    HP      HPC1533A
.
.
.
```

In this case, the ext_bus entry (which designates the bus adapter) specifies a scsi1 driver. You would configure the spt pass-through driver for the SCSI robotic controls on this system (see “Configuring Device Files for spt Pass-Through Driver” on page 95).

Example 2: Built-in SCSI interface (sctl driver)

```
ioscan -f
```

```
Class      I  H/W Path     Driver      S/W State H/W Type  Description
=============================================================================
ext_bus    2  10/12/5      c700        CLAIMED   INTERFACE Built-in SCSI
target    11  10/12/5.0    tgt         CLAIMED   DEVICE
tape       0  10/12/5.0.0  stape       CLAIMED   DEVICE    HP      C1533A
target    12  10/12/5.2    tgt         CLAIMED   DEVICE
disk       6  10/12/5.2.0  sdisk       CLAIMED   DEVICE    TOSHIBA CD-ROM
.
.
```

In this case, the ext_bus entry specifies a c700 driver. You would configure the sctl pass-through driver for the SCSI robotic controls on this system (see “Configure Device Files for sctl Pass-Through Driver” on page 97).

Configuring Device Files for spt Pass-Through Driver

Use this procedure on HP9000-800 systems that have a 28655A SCSI interface and use the scsi1 bus-adapter driver.

Note: The HP-UX kernel has to be reconfigured to use the spt SCSI pass-through driver on HP9000-800 systems. Refer to the HP-UX `scsi_pt (7)` man page.

The device files for the spt driver have the following format:

`/dev/spt/c`*Controller*`t`*Target*`d`*Unit*

Where:

Controller is the Instance number of the controlling bus. The Instance value is displayed in `ioscan -f` output under the column I of the controller's entry.

Target is the SCSI ID of the robotic control.

Unit is the SCSI logical unit number (LUN) of the robot. This is usually 0.

You must create the device files for the spt driver manually, as they are not created automatically when the system boots.

The following steps describe how to create these device files. These steps are also documented in the `scsi_pt(7)` man page.

1. Install and configure the driver as described in the `scsi_pt(7)` man page.
2. Determine the character major number of the spt driver using `lsdev`:

   ```
   lsdev -d spt
   ```

3. Use the following commands to create the device file for the SCSI robot control as follows:

   ```
   mkdir /dev/spt
   mknod /dev/spt/name c major 0xiitl00
   ```

   Where:

   name is the device name as described above.

   major is the character major number (from the previous `lsdev` command).

   ii is two hexadecimal digits identifying the controlling bus interface card by its Instance number.

   t is one hexadecimal digit representing the SCSI ID of robotic control.

   l is one hexadecimal digit representing the SCSI LUN of the robotic control.

Example of Device File

If the robotic control for an HP Optical Disk Library(ODL) is on a secondary SCSI bus at SCSI ID 3, LUN 0, use the following steps to create the device file.

1. Use `ioscan -f` to get information on the SCSI bus and the robotic control.

```
ioscan -f
```

```
Class   I  H/W Path  Driver  S/W State H/W Type  Description
=============================================================
bc      0             root    CLAIMED   BUS_NEXUS
bc      1  56         bc      CLAIMED   BUS_NEXUS Bus Converter
ext_bus 1  56/16     scsi1   CLAIMED   INTERFACE HP 28655A -  SCSI Interface
target  4  56/16.3   target  CLAIMED   DEVICE
spt     0  56/16.3.0 spt     CLAIMED   DEVICE    HP      C1700T
.
.
.
```

The Instance number for the robot's SCSI bus is 1. It also confirms that the spt driver is attached to the optical robotic control at H/W Path 56/16.3.0.

2. Use `lsdev` to get the character major number for the spt driver.

```
lsdev -d spt
```

This shows that the character major number for the spt driver is 137.

```
Character     Block       Driver          Class
  137          -1         spt             spt
```

3. Create the `/dev/spt` directory if it has not already been created.

```
mkdir /dev/spt
```

4. Create the device file as follows:

```
mknod /dev/spt/c1t3d0 c 137 0x013000
```

This creates the `/dev/spt/c1t3d0` device file. Specify this file as the robot control path when configuring your device under Media Manager.

Configure Device Files for sctl Pass-Through Driver

Use this procedure on HP9000-800 D, K, T, and V series systems that have a built-in SCSI interface and also on other systems that use the c700 bus-adapter driver. You do not have to reconfigure the HP-UX kernel to use sctl pass-through driver on HP9000-700 systems, since the generic SCSI driver is part of basic HP-UX.

If the devices do not exist, you can make device files by using the `mknod` command (see the `scsi_ctl(7)` man page) as follows:

```
mkdir /dev/sctl
cd /dev/sctl
/etc/mknod c controller t target l lun c 203 0xiitl00
```

Where:

controller is the Instance number of the controlling bus. The Instance value is displayed in `ioscan -f` output under column I of the controller entry (ext_bus in the Class column).

target is the SCSI ID of the robotic control.

lun is the SCSI logical unit number and should be 0 for all robots, except DLT2700, DLT4700, HP C1560B, and a few other robots where *lun* must be 1.

ii are two hexadecimal digits that identify the controlling bus interface card by its Instance number (same as controller).

t is one hexadecimal digit representing the SCSI ID.

l is one hexadecimal digit representing the SCSI LUN.

Notes on Using `ioscan` With sctl Robots

- If the robot is a LUN 1 robot (DLT4700, HP C1560B, and so on) there is no entry in the `ioscan` output for the robot.
- If the robotic control has its own SCSI ID, it has an entry similar to the following:

```
Class      I  H/W Path     Driver      S/W State H/W Type  Description
===================================================================
unknown    -1  2/0/1.1.0    unknown     UNCLAIMED UNKNOWN   LAGO SYSLS-340L
```

The Class I and Driver fields may also have invalid information. In these instances, the robotics are correct, but the `ioscan` command is not able to deal with them.

Examples of Device Files

Example 1

If the robotic control for a HP C1560B autoloader is on a built-in SCSI bus at SCSI ID 0 and the LUN is 1 (LUN is always 1 for HP C1560B autoloaders), use the following steps to create the device file.

1. Do an `ioscan -f` to get information on the SCSI bus and the robotic control.

```
ioscan -f
```

```
Class      I  H/W Path     Driver      S/W State H/W Type  Description
=============================================================================
ext_bus    2  10/12/5      c700        CLAIMED   INTERFACE Built-in SCSI
target    11  10/12/5.0     tgt         CLAIMED   DEVICE
tape       0  10/12/5.0.0   stape       CLAIMED   DEVICE    HP       C1533A
target    12  10/12/5.2     tgt         CLAIMED   DEVICE
disk       6  10/12/5.2.0   sdisk       CLAIMED   DEVICE    TOSHIBA CD-ROM
.
.
```

2. The commands to create the device file are:

```
cd /dev/sctl
/etc/mknod c2t0l1 c 203 0x020100
```

This creates the following device file, which you specify to Media Manager:

```
/dev/sctl/c2t0l1
```

Example 2

Assume the robotic control for an Exabyte 10i tape stacker (TS8) is on a built-in SCSI bus at SCSI ID 3, LUN 0. Also assume that an `ioscan -f` verifies that the SCSI ID is 3 and shows that the Instance number for the robot's SCSI bus is 1.

The commands to create the device file follow:

```
cd /dev/sctl
/etc/mknod c1t3l0 c 203 0x013000
```

This creates the following device file, which you specify to Media Manager:

```
/dev/sctl/c1t3l0
```

Configuring RS-232 Robotic Controls

Read this topic if you plan to use a robotic storage device that is controlled through an RS-232 connection. Supported RS-232 robots include:

- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TLD - Tape Library DLT

See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with the above robot types.

Configuring RS-232 controlled devices consists of determining the Instance number of the controlling interface and the port that will be used.

Device names for RS-232 connections use the following format:

`/dev/tty`*`I`*`p`*`Port`*

Where:

I is the Instance number of the controlling interface. The Instance value is displayed in `ioscan` output in the I column.

Port is the port number.

You can determine the Instance number with the following command:

```
ioscan -f -C tty
```

In the following output the interface Instance, found under column I, is 0.

```
Class     I  H/W Path  Driver S/W State H/W Type  Description
===================================================================
tty       0  56/56     mux2 CLAIMED   INTERFACE MUX
```

Example of RS-232 Robotic Control Device File

If the RS-232 robotic control connects to port 5 of the mux interface and the Instance number of the interface from `ioscan` is 0, the device file that you specify to Media Manager follows:

```
/dev/tty0p5
```

Configuring Tape Drives

Using Berkeley Style Close

The examples in this section show *Berkeley-style close* for tape drives as indicated by the letter `b` after the compression specification. It is mandatory to specify Berkeley-style close for tape devices that you configure under Media Manager.

The terms *Berkeley-style close* and *AT&T style close* refer to where a tape is left logically positioned after a close operation (in relation to a tape mark). One style leaves an application logically positioned before a tape mark and the other leaves it after. Applications must assume where the tape is left after a close in order to establish the correct orientation the next time they do a tape-position or read operation. Some operating systems allow tape devices to be configured with either type of close. NetBackup assumes it is using Berkeley-style close on an HP9000-800.

Device Files

When adding tape drives to the Media Manager configuration, you need only specify a no rewind on close device file path. These device files are found in the `/dev/rmt` directory and have the following format:

`/dev/rmt/c`*Controller*`t`*Target*`d`*Unit*`Density`*`nb`

Where:

Controller is the Instance number of the controlling bus. The Instance value is displayed in `ioscan -f` output under the column I of the controllers entry (ext_bus in the Class column).

Target is the SCSI ID of the tape drive.

Unit is the SCSI logical unit number (LUN) of the drive. This is usually 0.

Density is the desired density of the tape drive. For example:

- `D8MM_8500C` for 8505 drives
- `DDS2C` for DDS2 compression drives

See the `mt(7)` man page for an explanation of the densities.

If the desired tape device file does not exist, you can create device files through `sam`, the system administration manager or with the following `mksf(1M)` command:

`mksf -C tape -H` *H/W Path* `-b` *density* `-u -n`

Where:

H/W Path is the H/W Path of the tape drive as specified by `ioscan`.

density is the density string to use for the drive as defined in the `mksf(1M)` and `mt(7)` man pages.

Example of No Rewind Device File

Assume that the desired 4-mm DDS2 compression tape drive is at SCSI ID 2 and `ioscan` shows the following:

```
ioscan -f
```

```
Class   I  H/W Path Driver S/W State   H/W Type     Description
=============================================================================
bc      0           root    CLAIMED    BUS_NEXUS
bc      1 56        bc      CLAIMED    BUS_NEXUS    Bus Converter
ext_bus 0 56/52     scsi1   CLAIMED    INTERFACE    HP 28655A-SCSI Interface
target  0 56/52.2   target  CLAIMED    DEVICE
tape    0 56/52.2.0 tape2   CLAIMED    DEVICE       HP   HPC1533A
.
.
.
```

The Instance number for the controlling bus is 0 and the H/W path for the tape drive is 56/52.2.0.

The command to create the device file for drive follows:

```
mksf -C tape -H 56/52.2.0 -b DDS2C -u -n
```

This creates the following device file, which you specify to Media Manager:

```
/dev/rmt/c0t2d0DDS2Cnb
```

Fast-Tape Positioning (locate-block)

Locate block is supported for most drive types in HP9000-800 for Fast/Wide GSC SCSI adapters. See the *NetBackup Release Notes - UNIX* for a list of drive types that are supported.

Note: Locate is not supported on HP-PB adapters such as HP 28696A - Wide SCSI or HP 28655A - SE SCSI.

To enable locate block on Fast/Wide GSC SCSI adapter, a device file in the directory `/dev/sctl` must exist for the **tape drives**. Create the device files as documented in “Configure Device Files for sctl Pass-Through Driver" on page 97.

Example:

Assume the configuration from `ioscan -f` is as follows:

```
Class    I  H/W Path  Driver  S/W State H/W Type   Description
=============================================================
ext_bus 0  10/0     c720   CLAIMED  INTERFACE GSC built-in Fast/Wide
 SCSI Interface
tape    5  10/0.1.0 stape  CLAIMED  DEVICE    Quantum DLT4000
tape    6  10/0.2.0 stape  CLAIMED  DEVICE    Quantum DLT4000
.
.
```

The tape drives are SCSI IDs 1 and 2 on ext_bus 0. In the above example, the robotics for the robot is SCSI ID 0 (it does not show up with `ioscan`). In the directory `/dev/sctl`, the following device files were created as shown:

```
# cd /dev/sctl
# ls -l

total 0
crw-r--r--   1 root     sys      203 0x000000 Jun 24 14:19 c0t0l0
crw-r--r--   1 root     sys      203 0x001000 Jun 24 14:20 c0t1l0
crw-rw-rw-   1 root     sys      203 0x002000 Mar 27 12:46 c0t2l0
```

The first one is used for the SCSI robotics. The second two are created for use by NetBackup and Storage Migrator to perform locate block on the tape drives. These device files have to exist, but are not used for any configuration in Media Manager. They **must** be of the form `c`*Adapter*`t`*Target*`l`*Lun*.

To disable locate block (once it is enabled), remove the `/dev/sctl` device file created for the tape drive.

Switch Settings for HP C1533A 4-mm DAT Drives

If you have standalone or robotic 4-mm drives, model HP C1533A, you may have to change the switch settings on the bottom of the drive. This is the drive that comes in the HP C1560B (48AL) DAT Autoloader.

If the C1533A drive or HP C1560B autoloader was purchased from Hewlett Packard, the default switch settings should work. These default settings, as documented by HP, are as follows:

On=1, Off=0

Switch	Setting
1	1
2	1
3	0
4	1
5	1
6	1
7	1
8	1

However, if the drive or autoloader were purchased from another vendor, and that vendor changed the switch settings, you will have to set the switches as shown above.

You may also have to make this change to HP C1533A drives in non-Hewlett Packard 4-mm robots.

Configuring Optical Disk Drives

When adding optical disk drives to the Media Manager configuration, you need only specify a character device path. Optical disk character device files are found in the `/dev/rdsk` directory, and have the following format:

`/dev/rdsk/c`*`BI`*`t`*`Target`*`d0`

Where:

BI is the bus Instance number of the controlling bus. The Instance value is displayed in `ioscan` output under the column I of the ext_bus entries.

Target is the SCSI ID of the drive. This ID is in the third position of the H/W Path as displayed by `ioscan`. For example, in 56/52.5.0 the SCSI ID is 5.

You can determine the bus Instance with the following command:

```
ioscan -C ext_bus -f
```

```
Class   I  H/W Path Driver S/W State H/W Type   Description
=============================================================
ext_bus 0  56/52    scsi1  CLAIMED   INTERFACE  HP 28655A
 - SCSI Interface
ext_bus 1  56/53    lpr2   CLAIMED   INTERFACE  HP 28655A
 - Parallel Interface
```

You can determine the configured drives with the following command:

```
ioscan -C disk -f
```

```
Class   I  H/W Path  Driver S/W State H/W Type  Description
=============================================================
disk    1  56/52.1.0 disc3  CLAIMED   DEVICE    HP C1716T
disk    2  56/52.2.0 disc3  CLAIMED   DEVICE    HP C1716T
disk    3  56/52.5.0 disc3  CLAIMED   DEVICE    HP C2490AM
disk    4  56/52.6.0 disc3  CLAIMED   DEVICE    HP C2490AM
```

Example of an Optical Disk Device File

Assume you are using the two optical disk drives at SCSI IDs 1 and 2 as shown in the disk `ioscan` example above. These drives are on bus 56/52, which as shown in the ext_bus `ioscan` above, is bus Instance 0.

The character device file paths that you specify to Media Manager follow:

For target 1:

```
/dev/rdsk/c0t1d0
```

For target 2:

```
/dev/rdsk/c0t2d0
```

Command Summary

The following is a summary of commands that may be useful when configuring devices. See the procedures in this chapter for examples of usage.

`ioscan -C` *type* `-f`

Shows information about the physical interfaces. *type* is the type of interface as follows:
`spt` specifies SCSI robotic controls.
`tty` specifies RS-232 robotic controls.
`tape` specifies tape drives.
`disk` specifies optical disks.
`ext_bus` specifies SCSI controllers.

Note: Numeric information is displayed in decimal.

`mknod /dev/spt/`*name* `c` *major* `0x`*ii**t**l*`00`

Creates device files for SCSI robotic controls.
name is the device name as described in the format: c *controller* t *target* d *unit*.
major is the character major number (from `lsdev`).
ii are the two hexadecimal digits identifying the controlling bus interface card by its Instance number. The Instance value is displayed in the `ioscan` output under the I column of the proper ext_bus entry.
t is one hexadecimal digit for the SCSI ID of the robotic control.
l is one hexadecimal digit for the SCSI LUN of the robotic control.

`lsdev -d spt`

Displays information about the SCSI robotic control drivers.

`mksf -C tape -H` *H/W Path* `-b` *density* `-u -n`

Creates device files for tape drives. Where *H/W Path* is the hardware path of the tape drive as specified by `ioscan` and *density* is the density string to use for the drive as defined in the `mksf(1M)` and `mt(7)` man pages.

IRIX 6.2

This chapter provides information for configuring devices for use with Media Manager on an SGI platform running IRIX. You configure drives and robots using one of the available Media Manager administrative interfaces.

Typical device path names used when configuring drives and robots are described in this chapter. Instructions for changing and rebuilding the kernel are also included. Depending on the type and number of devices you are adding, you may have to enter information into kernel source files and then reconfigure the kernel.

The topics included in this chapter are as follows:

- Using SCIP Controllers
- Note on the mediad Command
- Configuring Robotic Controls
- Configuring Tape Drives
- Configuring Optical Disk Drives
- Command Summary

To determine which drive and robot types are supported on IRIX, refer to the *NetBackup Release Notes - UNIX*.

Be sure to observe the following points when performing the configuration described in this chapter:

- The SGI IRIX version of Media Manager has been tested using SCSI peripherals (tape drives, optical disk drives, and robotic control) attached to the built-in SCSI controllers, sometimes referred to as on-board SCSI or Integral SCSI controllers. When referring to these SCSI controllers, this chapter uses the term *integral SCSI controller*. Communication with tape drives attached to integral SCSI controllers is done through the `tps(7M)` tape driver. Communication with disk drives (including optical disk drives) attached to integral SCSI controllers is done through the `dks(7M)` disk driver.
- VERITAS does not recommend or support the use of single-ended to differential SCSI converters on Media Manager controlled devices. You may encounter problems if you use these converters.
- You can copy code examples from the on-line version of this manual, located in `/usr/openv/volmgr/dev_config_guide.txt`. This file is installed with NetBackup and Media Manager software.

Using SCIP Controllers

If your IRIX system has SCIP fast-wide-differential controllers, a change to the `/var/sysgen/master.d/scip` file may be required to avoid SCSI time-outs.

You should change the following:

```
uint                            scip_mintimeout = 0
```

To

```
uint                            scip_mintimeout = 180
```

The above value was tested with a Quantum DLT4700 and corrected driver errors. In general, it's better to try a peripheral first without modifying this file. If errors occur, then change the timeout and retry. You may have to contact Silicon Graphics Corporation for further information.

After making this change, you must generate a new kernel and reboot the system as follows:

1. Run the kernel auto-configuration script:

   ```
   /etc/autoconfig
   ```

2. Reboot the system to utilize the newly built kernel.

Note on the `mediad` Command

Do not use the IRIX `mediad` command to monitor devices configured under Media Manager. If you do, Media Manager will not be able to access the devices and you will see a message similar to the following in the system log:

```
Apr 12 10:30:55 3D:boris mediad: Could not access
device /dev/rmt/tps0d4nr, Device busy
```

If you see this type of message and you are using `mediad`, then, disable `mediad` as described in the `mediad(1M)` man page.

For example, assume you encounter this problem with a tape device whose device file is `/dev/rmt/tps0d4`. Here, you could execute the following sequence of commands:

1. Instruct `mediad` to not monitor this tape device by executing:

   ```
   /usr/etc/mediad -r /dev/rmt/tps0d4
   ```

   This does not take effect until the next time you start `mediad`. But, it immediately adds an entry to the `/etc/fsd.auto` file, which you can verify with the `more` command:

   ```
   more /etc/fsd.auto

   /dev/rmt/tps0d1 /tape mediad rw,mon=off 0 0
   /dev/rmt/tps0d5 /tape mediad rw,mon=off 0 0
   /dev/rmt/tps1d6 /tape mediad rw,mon=off 0 0
   /dev/rmt/tps1d7 /tape mediad rw,mon=off 0 0
   /dev/rmt/tps0d4 /tape mediad rw,mon=off 0 0
   ```

2. Stop `mediad` as follows:

   ```
   /usr/etc/mediad -k
   ```

3. Restart `mediad` as follows:

   ```
   /usr/etc/mediad
   ```

Configuring Robotic Controls

Robots are controlled through a SCSI, an RS-232, or a network connection. Configuration for network controlled robotic libraries is discussed in the appendices of the *Media Manager System Administrator's Guide - UNIX*. SCSI and RS-232 control is covered in the following sections.

Configuring SCSI Robotic Controls

Read this topic if you plan to use a robotic device that is controlled through a SCSI robotic connection. Supported SCSI robots include the following:

- ODL - Optical Disk Library
- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TL4 - Tape Library 4MM
- TL8 - Tape Library 8MM
- TLD - Tape Library DLT
- TS8 - Tape Stacker 8MM
- TSD - Tape Stacker DLT
- TSH - Tape Stacker Half-inch

See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with the above robot types.

When communicating with SCSI-controlled robotic peripherals on an SGI computer, Media Manager robotic software utilizes `ds(7M)`, the generic (user mode) SCSI driver. Since this driver is part of basic IRIX, you do not have to reconfigure the kernel and reboot the system to use this driver.

You can use the `/dev/MAKEDEV` script to create the correct device files for the generic SCSI driver. Use the following procedure:

1. Before running the script for a DLT2700/DLT4700), HP C1560B, and few other robot types, remember these devices have a logical unit number of 1 and need the following change:

   Modify the `/dev/MAKEDEV.d/SCSI_base` file as follows.

Find the line containing

```
for lun in 0; do \
```

and make the following change:

```
for lun in 0 1; do \
```

2. Create all the device files for the generic SCSI driver by executing the following commands:

```
cd /dev
./MAKEDEV scsi
```

The device files for the SCSI controlled robotics reside in the `/dev/scsi` directory and the format of these files is:

`/dev/scsi/sc`*Controller*`d`*Target*`l`*Lun*

Where:

Controller is the SCSI bus (adapter) number.

Target is the SCSI ID.

Lun is the logical unit number.

Examples of SCSI Robotic Control Device Files

Note: The second-to-last character in the following example device file paths is the letter l, rather than the number 1 and stands for logical unit.

Example 1

If the robotic control is not for a DLT2700, DLT4700, HP C1560B, and few other robot types and the control is on SCSI bus (adapter) 0 at SCSI ID 5, the path you enter is:

```
/dev/scsi/sc0d5l0
```

Example 2

If the robotic control is not for a DLT2700, DLT4700, HP C1560B, and few other robot types and the control is on SCSI bus (adapter) 1 at SCSI ID 3, the path you use follows:

```
/dev/scsi/sc1d3l0
```

Example 3

If the robotic control is for a DLT2700, DLT4700, HP C1560B, and a few other robot types and the control is on SCSI bus (adapter) 1 at SCSI ID 4 and logical unit number 1, the device path you use follows:

```
/dev/scsi/sc1d4l1
```

Configuring RS-232 Robotic Controls

Read this topic if you plan to use a robotic device that is controlled through an RS-232 connection. Supported RS-232 robots include:

- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TLD - Tape Library DLT

See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with the above robot types.

Configuring RS-232 Controlled devices on an IRIX platform normally consists of determining the port number of the serial ports being used.

Examples of RS-232 Robotic Control Device Files

The path names that you use when adding the RS-232 controlled robotic device to your configuration have the following format:

`/dev/ttyf`*`NN`*

Where *NN* is the port number.

For example, if the RS-232 (serial) control line is attached to port 3, the path you use follows:

```
/dev/ttyf3
```

Configuring Tape Drives

Fast-Tape Positioning (locate-block)

For most drive types, Media Manager supports the SCSI `locate-block` command for positioning a tape to a specific block. This improves tape-positioning times over the alternative method, which is the forward-space-file / record method. See the *NetBackup Release Notes - UNIX* for a list of drive types that support `locate-block`.

NetBackup and Storage Migrator use the `locate-block` command by default unless you disable it by executing:

```
touch /usr/openv/volmgr/database/NO_LOCATEBLOCK
```

With locate-block positioning disabled, NetBackup uses the forward-space-file / record method and Storage Migrator skips file marks.

Creating Device Files

CAUTION! In this section, the device file examples for non-QIC drives specify variable mode as indicated by a `v` after the no rewind on close specification. You must configure non-QIC tape drives as variable-mode devices if they will be used by NetBackup on SGI IRIX platforms. If these devices are not variable mode, NetBackup is able to write data but cannot read it. During a read, you see a "not in tar format" error. The terms *variable mode* and *fixed mode* refer to the behavior of reads and writes and the way the kernel packs physical tape records into logical tape records for an application. Variable-mode devices allow more flexibility in reading previously written tapes. Many tape devices can be accessed in either mode. NetBackup assumes variable mode for non-QIC drives. See `MAKEDEV(1M)` and `tps(7M)` for more information.

When adding tape drives to a Media Manager configuration, you need only specify a no rewind on close device path. In a typical configuration, most of the desired tape device files exist and you just have to locate them in the `/dev` directory.

No rewind on close device files that connect to the integral SCSI controllers have the following format:

`/dev/rmt/tps`*Controller*`d`*Target*`nrv`

Where:

Controller is the SCSI bus (adapter) number

Target is the SCSI ID

The `v` specifies a variable mode device.

Some device types (like Exabyte) also have suffixes on their device files that designate their particular drive type. For example:

`/dev/rmt/tps`*Controller*`d`*Target*`nrv.8500c` (EXB8500C)

If the device files you need do not exist, you can use the `MAKEDEV` command to create all the tape device file combinations for `tps` (the SCSI tape driver for Integral SCSI controllers) as follows:

```
cd /dev
./MAKEDEV tps
```

Examples of No Rewind Device Files

Example 1

If the desired HP 4-mm (DAT) drive is on SCSI bus 1 at SCSI ID 4, you specify the following device path for that drive:

`/dev/rmt/tps1d4nrv`

Example 2

If the desired Exabyte 8500C or 8505 tape drive is on SCSI bus 0 at SCSI ID 3, you specify the following device path for that drive:

`/dev/rmt/tps0d3nrv.8500c`

Example 3

If the desired DLT tape drive is on SCSI bus 0 at SCSI ID 5, you specify the following device file path:

`/dev/rmt/tps0d5nrvc`

Example 4

If the desired Exabyte 8900 (Mammoth) is on SCSI bus 1 at SCSI ID 5, you specify the following device path for that drive:

```
/dev/rmt/tps1d5nrv
```

Since this drive writes in only one format, you can ignore other device files that are created for this drive.

Adding Exabyte Compression Drives

If you plan to use standalone or robotic Exabyte compression drives (8500C, 8505, 8505XL, and 8900), the `struct tpsc_types tpsc_types[]` array must contain entries for these drives. These entries are needed for an IRIX system to recognize the drives. You can find this array in the `/var/sysgen/master.d/scsi` file.

Note: The entries for the 8500C, 8505 and 8505XL drives are in the released version of this file. You have to add an entry only if you are using an 8900 (Mammoth) drive.

Changing `/var/sysgen/master.d/scsi`

Add an entry for an Exabyte 8900 drive as follows:

1. If the following code is in `/var/sysgen/master.d/scsi` and you have previously rebuilt the kernel and changed `MAKEDEV` as explained in "Reconfiguring the Kernel and Modifying MAKEDEV", then no further changes are necessary.

```
/*
* Exabyte 8900 (Mammoth) drives.
*/
{ EXABYTE8900, TP8MM_8900, 7, 8, "EXABYTE", "EXB-8900", 6,
"\40\4\16\0\200\7", {0x27, 0x27, 0x27, 0x00},
MTCAN_BSF|MTCAN_BSR|MTCAN_PREV|MTCAN_CHKRDY|MTCAN_VAR|MTCAN_SETSZ|
MTCAN_SILI|MTCAN_CHTYPEANY|MTCAN_SETDEN|MTCAN_SPEOD|MTCAN_SYNC|
MTCAN_SEEK, 80, 4*60, 25*60, 5*60, 1024, 128*1024, 0, (u_char*)0
},
```

2. If the code is not in the file, add it as follows:

   a. Save a copy of `/var/sysgen/master.d/scsi`.

   b. Add the code. An easy way to make this addition is to copy the code from the `dev_config_guide.txt` file.

c. Reconfigure the kernel as explained in “Reconfiguring the Kernel and Modifying MAKEDEV”.

Reconfiguring the Kernel and Modifying MAKEDEV

If you made any changes to the `/var/sysgen/master.d/scsi` file, then you must reconfigure the kernel and modify the `MAKEDEV` script as follows:

1. Run the following kernel auto-configuration script:

   ```
   /etc/autoconfig
   ```

2. Reboot the system to utilize the newly built kernel.

Adding HP 4-mm Drives and HP C1560B DAT Autoloaders

Read these topics if you plan to use standalone or robotic Hewlett-Packard 4-mm DAT tape drives or HP C1560B DAT Autoloaders. They explain drive switch settings and kernel changes you may have to make in order for the system to recognize these devices.

Checking Switch Settings

Ensure that the hardware (tape drive) switch settings on HP35480A 4-mm (DAT) drives are as follows:

Note: Other combinations may work, but these are the settings that were functional during testing by VERITAS with an HP35480A drive and HP C1560B Autoloader.

On=1, Off=0

Switch	Setting
1	1
2	1
3	1
4	1
5	1
6	1
7	0
8	0

Ensure that the hardware (tape drive) switch settings on the HP C1533A 4-mm (DAT) drives are as follows:

Switch	Setting
1	1
2	1
3	0
4	1
5	1
6	1
7	0
8	0

Changing /var/sysgen/master.d/scsi

For the system to recognize the 4-mm DAT drives, the `struct tpsc_types tpsc_types[]` array must have code entries for the drives. You will find this array in the `/var/sysgen/master.d/scsi` file.

1. The code entries that must be in this array are as follows:

   For all DAT drives except an HP C1560B DAT Autoloader, the following entry must exist:

```
/*
 * HP DAT drives. Any product number that starts with HP354.
 */
 { DATTAPE, TPDAT, 2, 5, "HP", "HP354", 0, 0, {0, 0, 0, 0},
MTCAN_BSF|MTCAN_BSR|MTCAN_APPEND|MTCAN_SETMK|MTCAN_PART|MTCAN_PREV|
MTCAN_SYNC|MTCAN_SPEOD|MTCAN_CHKRDY|MTCAN_VAR|MTCAN_SETSZ|
MTCAN_SILI|MTCAN_SEEK|MTCAN_CHTYPEANY,
 /* minimum delay on i/o is 4 minutes, because when a retry is
 * performed, the drive retries a number of times, and then
 * rewinds to BOT, repositions, and tries again.  */
 40, 4*60, 4*60, 5*60, 512, 128*512, 0, (u_char*)0
 },
```

   For an HP C1560B DAT Autoloader, add the following entry:

```
/*
 * HP DAT drives. Any product number that starts with HPC1533.
 */
 { DATTAPE, TPDAT, 2, 5, "HP", "C1533", 0, 0, {0, 0, 0, 0},
MTCAN_BSF|MTCAN_BSR|MTCAN_APPEND|MTCAN_SETMK|MTCAN_PART|MTCAN_PREV|
MTCAN_SYNC|MTCAN_SPEOD|MTCAN_CHKRDY|MTCAN_VAR|MTCAN_SETSZ|
MTCAN_SILI|MTCAN_SEEK|MTCAN_CHTYPEANY,
 /* minimum delay on i/o is 4 minutes, because when a retry is
 * performed, the drive retries a number of times, and then
 * rewinds to BOT, repositions, and tries again.  */
 40, 4*60, 4*60, 5*60, 512, 128*512, 0, (u_char*)0
 },
```

2. If the above code is in `/var/sysgen/master.d/scsi` and you have previously rebuilt the kernel as explained in Step c of Step 3 below, then no further changes are necessary.

3. If the above code is not in `/var/sysgen/master.d/scsi`, then add it as follows:

   a. Save a copy of the `/var/sysgen/master.d/scsi` file.

   b. Add the code. The easiest way to make this addition is to copy it from the `dev_config_guide.txt` file.

   c. After completing your changes to the `/var/sysgen/master.d/scsi` file, reconfigure the kernel by running the kernel auto-configuration script:

      ```
      /etc/autoconfig
      ```

   d. Reboot the system to utilize the newly built kernel.

Adding Quantum DLT Drives or Stackers

Read this topic if you plan to use standalone or robotic Quantum DLT2000, DLT4000 or DLT7000 tape drives or Quantum DLT2700 or DLT4700 stackers.

For the system to recognize the DLT drives, the `struct tpsc_types tpsc_types[]` array must contain entries for them. You will find this array in the `/var/sysgen/master.d/scsi` file.

1. The code entries that must be in this array are as follows:

   For an older DLT2000 drive that uses DEC for the vendor ID:

   ```
   /* DEC  DLT2000 drive */
   { DECDLT, TPDLT, 3, 7, "DEC", "DLT2000", 0, 0,
      {0, 0, 0, 0}, MTCAN_BSF|MTCAN_BSR|MTCAN_APPEND|MTCAN_LEOD|
      MTCAN_CHKRDY|MTCAN_VAR|MTCAN_SETSZ|MTCAN_SILI|MTCAN_SEEK|
      MTCAN_SYNC|MTCAN_CHTYPEANY|MTCAN_COMPRESS,
     20, 8*60, 20*60, 5*60, 4096, 64*1024, 0, (u_char*)0 },
   ```

   For an older DLT2700 stacker that uses DEC for the vendor ID:

   ```
   /* DEC DLT2700 Stacker*/
   { DECDLT, TPDLT, 3, 7, "DEC", "DLT2700", 0, 0,
      {0, 0, 0, 0}, MTCAN_BSF|MTCAN_BSR|MTCAN_APPEND|MTCAN_LEOD|
      MTCAN_CHKRDY|MTCAN_VAR|MTCAN_SETSZ|MTCAN_SILI|MTCAN_SEEK|
      MTCAN_SYNC|MTCAN_CHTYPEANY|MTCAN_COMPRESS,
      20, 8*60, 20*60, 5*60, 4096, 64*1024, 0, (u_char*)0 },
   ```

For a Quantum DLT2000 drive (for a Quantum DLT2700 stacker, substitute DLT2700 for DLT2000):

```
/* QUANTUM  DLT2000 drive */
{ DECDLT, TPDLT, 7, 7, "Quantum", "DLT2000", 0, 0,
   {0, 0, 0, 0}, MTCAN_BSF|MTCAN_BSR|MTCAN_APPEND|MTCAN_LEOD|
   MTCAN_CHKRDY|MTCAN_VAR|MTCAN_SETSZ|MTCAN_SILI|MTCAN_SEEK|
   MTCAN_SYNC|MTCAN_CHTYPEANY|MTCAN_COMPRESS,
   20, 8*60, 20*60, 5*60, 4096, 64*1024, 0, (u_char*)0 },
```

For a Quantum DLT4000 drive (for a Quantum DLT4700 stacker, substitute DLT4700 for DLT4000):

```
/* DEC DLT4000 Stacker*/
{ DECDLT, TPDLT, 7, 7, "Quantum", "DLT4000", 0, 0,
   {0, 0, 0, 0}, MTCAN_BSF|MTCAN_BSR|MTCAN_APPEND|MTCAN_LEOD|
   MTCAN_CHKRDY|MTCAN_VAR|MTCAN_SETSZ|MTCAN_SILI|MTCAN_SEEK|
   MTCAN_SYNC|MTCAN_CHTYPEANY|MTCAN_COMPRESS,
   20, 8*60, 20*60, 5*60, 4096, 64*1024, 0, (u_char*)0 },
```

For a Quantum DLT7000 drive:

```
/* DEC DLT7000 drive */
{ DECDLT, TPDLT, 7, 7, "QUANTUM", "DLT7000", 0, 0,
   {0, 0, 0, 0}, MTCAN_BSF|MTCAN_BSR|MTCAN_APPEND|MTCAN_LEOD|
   MTCAN_CHKRDY|MTCAN_VAR|MTCAN_SETSZ|MTCAN_SILI|MTCAN_SEEK|
   MTCAN_SYNC|MTCAN_CHTYPEANY|MTCAN_COMPRESS,
   20, 8*60, 20*60, 5*60, 4096, 64*1024, 0, (u_char*)0 },
```

2. If the above code is in `/var/sysgen/master.d/scsi` and you have previously rebuilt the kernel as explained in Step c of Step 3 below, then no further changes are necessary.

3. If the code is not in `/var/sysgen/master.d/scsi`, then add it as follows:

   a. Save a copy of `/var/sysgen/master.d/scsi`.

   b. Add the above code. The easiest way to make this addition is to copy it from the `dev_config_guide.txt` file.

   c. After completing your changes to the `/var/sysgen/master.d/scsi` file, reconfigure the kernel by running the following kernel auto-configuration script:

      `/etc/autoconfig`

   d. Reboot the system to utilize the newly built kernel.

Adding Sony DTF Drives

For the system to recognize the DTF cartridge tape drives, the struct `tpsc_types tpsc_types[]` array must contain entries for them. You will find this array in the `/var/sysgen/master.d/scsi` file.

1. The code entry that must be in this array follows:

```
/*  Sony DTF drive */
   { DECDLT, TPDLT, 4, 7, "SONY", "GY-2120", 0, 0,
   (0}, MTCAN_BSF|MTCAN_BSR|MTCAN_APPEND|MTCAN_SPEOD|
   MTCAN_CHKRDY|MTCAN_VAR|MTCAN_SETSZ|MTCAN_SILI|MTCAN_SEEK|
   MTCAN_SYNC|MTCAN_CHTYPEANY|MTCAN_COMPRESS,
   20, 8*60, 20*60, 5*60, 4096, 64*1024, 0, (u_char *)0 },
```

2. If the above code is in `/var/sysgen/master.d/scsi` and you have previously rebuilt the kernel as explained in Step c of Step 3 below, then no further changes are necessary.

3. If the code is not in `/var/sysgen/master.d/scsi`, add it as follows:

   a. Save a copy of `/var/sysgen/master.d/scsi`.

   b. Add the above code. The easiest way to make this addition is to copy it from the `dev_config_guide.txt` file.

   c. After completing your changes to the `/var/sysgen/master.d/scsi` file, reconfigure the kernel by running the kernel auto-configuration script:

      `/etc/autoconfig`

   d. Reboot the system to utilize the newly built kernel.

Configuring Optical Disk Drives

When adding optical disk drives to a Media Manager configuration, specify the following device paths:

- Character device path (disk partition `s7`)
- Volume header disk device path (disk partition `vh`)

In a typical SGI IRIX configuration, most of the desired disk device files exist and you just have to locate them in the `/dev` directory.

Character disk device files have the following format:

`/dev/rdsk/dks`*Controller*`d`*Target*`s7`

Volume disk device files have the following format:

`/dev/rdsk/dks`*Controller*`d`*Target*`vh`

Where:

Controller is the SCSI bus (adapter) number.

Target is the SCSI ID.

`s7` is the desired disk partition.

`vh` is the desired disk partition.

If the required device files do not exist, you can create all the disk device file combinations for `dks` (the SCSI disk driver for integral SCSI controllers) by using the `MAKEDEV` command as follows:

```
cd /dev
./MAKEDEV dks
```

Example of Optical Disk Device Files

If the desired optical disk drive is on SCSI bus 1 at SCSI ID 3, you specify the following paths:

`/dev/rdsk/dks1d3vh` (volume header)
`/dev/rdsk/dks1d3s7` (character device)

Command Summary

The following is a summary of commands that may be useful when configuring devices. See the procedures in this chapter for examples of their usage.

`/usr/etc/mediad -r /dev/rmt/`*device*

`/usr/etc/mediad -k`

`/usr/etc/mediad`
: Use the `-r` option to prevent `mediad` from monitoring Media Manager controlled tape devices (where *device* specifies the device file for the tape). Media Manager is unable to access tape devices that are being monitored by `mediad`.
This command does not take effect until the next time you start `mediad`. But, it immediately adds an entry to the `/etc/fsd.auto` file, which you can verify with the `more` command. Use the `-k` option to stop `mediad` and restart it by executing the command with no options.

`MAKEDEV` *type*
: If the device files you need do not exist, you can execute the `MAKEDEV` command from the `/dev` directory to create them.
type indicates the type of device file as follows:
`tps` creates all the tape device file combinations for `tps` (the SCSI tape driver for Integral SCSI controllers).
`scsi` creates all the device files for the generic SCSI driver.
`dks` creates all the device files for dks (the SCSI disk driver for integral SCSI controllers).

`/etc/autoconfig`
: Runs the kernel auto-configuration script.

`/usr/openv/volmgr/bin/vmconf`
: Provided with Media Manager, this script eases device setup in less complex configurations.

`/sbin/hinv`
: Shows the system configuration, including devices configured on SCSI controllers.

NetBackup 3.2 Media Manager Device Configuration Guide for UNIX

IRIX 6.4/6.5

This chapter provides information for configuring devices for use with Media Manager on an SGI platform running IRIX. You configure drives and robots using one of the available Media Manager administrative interfaces.

Typical device path names used when configuring drives and robots are described in this chapter. Instructions for changing and rebuilding the kernel are also included. Depending on the type and number of devices you are adding, you may have to enter information in kernel source files and then reconfigure the kernel.

The topics included in this chapter are as follows:

- Using SCIP Controllers
- Note on the mediad Command
- Configuring Robotic Controls
- Configuring Tape Drives
- Configuring Optical Disk Drives
- Command Summary

To determine which drive and robot types are supported, refer to the *NetBackup Release Notes - UNIX*.

Be sure to observe the following points when performing the configuration described in this chapter:

- The SGI IRIX version of Media Manager has been tested using SCSI peripherals (tape drives, optical disk drives, and robotic control) attached to the built-in SCSI controllers, sometimes referred to as on-board SCSI or Integral SCSI controllers. When referring to these SCSI controllers, this document uses the term *integral SCSI controller*. Communication with tape drives attached to integral SCSI controllers is done through the `tps(7M)` tape driver. Communication with disk drives (including optical disk drives) attached to integral SCSI controllers is done through the `dks(7M)` disk driver.
- VERITAS does not recommend or support the use of single-ended to differential SCSI converters on Media Manager controlled devices. You may encounter problems if you use these converters.
- You can copy code examples from the on-line version of this manual, located in `/usr/openv/volmgr/dev_config_guide.txt`. This file is installed with Media Manager software.

Using SCIP Controllers

If your IRIX system has SCIP fast-wide-differential controllers, a change to the `/var/sysgen/master.d/scip` file may be required to avoid SCSI timeouts.

Change:

```
uint                            scip_mintimeout = 0
```

To:

```
uint                            scip_mintimeout = 180
```

The above change was tested with a Quantum DLT4700 and corrected driver errors. In general, it's better to try a peripheral first without modifying this file. If errors occur, then change the timeout and retry. You may have to contact Silicon Graphics Corporation for further information.

After making this change, you must generate a new kernel and reboot the system as follows:

1. Run the following kernel auto-configuration script:

   ```
   /etc/autoconfig
   ```

2. Reboot the system to utilize the newly built kernel.

Note on the `mediad` Command

Do not use the IRIX `mediad` command to monitor devices configured under Media Manager. If you do, Media Manager will not be able to access the devices and you will see a message similar to the following in the system log:

```
Apr 12 10:30:55 3D:boris mediad: Could not access
device /dev/rmt/tps0d4nr, Device busy
```

If you see this type of message and you are using `mediad`, then, disable `mediad` as described in the `mediad(1M)` man page.

For example, assume you encounter this problem with a tape device whose device file is `/dev/rmt/tps0d4`. Here, you could execute the following sequence of commands:

1. Instruct `mediad` to not monitor this tape device by executing:

   ```
   /usr/etc/mediad -r /dev/rmt/tps0d4
   ```

   This does not take effect until the next time you start `mediad`. But, it immediately adds an entry to the `/etc/fsd.auto` file, which you can verify with the `more` command:

   ```
   more /etc/fsd.auto

   /dev/rmt/tps0d1 /tape mediad rw,mon=off 0 0
   /dev/rmt/tps0d5 /tape mediad rw,mon=off 0 0
   /dev/rmt/tps1d6 /tape mediad rw,mon=off 0 0
   /dev/rmt/tps1d7 /tape mediad rw,mon=off 0 0
   /dev/rmt/tps0d4 /tape mediad rw,mon=off 0 0
   ```

2. Stop `mediad` as follows:

   ```
   /usr/etc/mediad -k
   ```

3. Restart `mediad` as follows:

   ```
   /usr/etc/mediad
   ```

Configuring Robotic Controls

Robots are controlled through a SCSI, an RS-232, or a network connection. Configuration for network controlled robotic libraries is explained in the appendices of the *Media Manager System Administrator's Guide - UNIX*. SCSI and RS-232 control is covered in the following sections.

Configuring SCSI Robotic Controls

Read this topic if you plan to use a robotic device that is controlled through a SCSI robotic connection. Supported SCSI robots include the following:

- ODL - Optical Disk Library
- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TL4 - Tape Library 4MM
- TL8 - Tape Library 8MM
- TLD - Tape Library DLT
- TS8 - Tape Stacker 8MM
- TSD - Tape Stacker DLT
- TSH - Tape Stacker Half-inch

See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with the above robot types.

When communicating with SCSI-controlled robotic peripherals on an SGI platform, Media Manager robotic software utilizes `ds(7M)`, the generic (user mode) SCSI driver. Since this driver is part of basic IRIX, you do not have to reconfigure the kernel and reboot the system to use this driver.

Examples of SCSI Robot Control Device Files

Note: Note that the second-to-last character in the following example paths is the letter l, rather than the number 1 and stands for logical unit.

Example 1

If the robotics control is not for a DLT2700, DLT4700, HP C1560B, or other LUN 1 peripheral and is on SCSI bus (adapter) 0 at SCSI ID 5, the device file you specify is:

```
/dev/scsi/sc0d5l0
```

Example 2

If the robotics control is not for a DLT2700, DLT4700, HP C1560B, or other LUN 1 peripheral and is on SCSI bus (adapter) 1 at SCSI ID 3, the device file you specify is:

```
/dev/scsi/sc1d3l0
```

Example 3

If a DLT2700, DLT4700, HP C1560B, or other LUN 1 peripheral robotics control is on SCSI bus (adapter) 1 at SCSI ID 4 with logical unit number 1, the device file you specify is:

```
/dev/scsi/sc1d4l1
```

Configuring RS-232 Robotic Controls

Read this topic if you plan to use a robotic device that is controlled through an RS-232 connection. See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with the following supported RS-232 robot types:

- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TLD - Tape Library DLT

Configuring RS-232 Controlled devices on an SGI IRIX platform normally consists of determining the port number of the serial ports being used.

Examples of RS-232 Robot Control Device Files

The path names that you use when adding a RS-232 controlled robotic device to your configuration have the following format:

`/dev/ttyf`*`NN`*

Where *NN* represents the port number.

For example, if the RS-232 (serial) control line is attached to port 3, the path you enter is as follows:

```
/dev/ttyf3
```

Configuring Tape Drives

Read the following topics if you plan to use tape drives. With locate-block positioning disabled, NetBackup uses the forward-space-file/record method.

When adding tape drives to a Media Manager configuration, you need only specify a no rewind on close device path. In a typical configuration, most of the desired tape device files exist and you just have to locate them in the `/dev` directory.

No rewind on close device files that connect to the integral SCSI controllers have the following format:

`/dev/rmt/tps`*Controller*`d`*Target*`nrv`

Where:

Controller is the SCSI bus (adapter) number

Target is the SCSI ID

The `v` specifies a variable mode device.

Some device types (like Exabyte) also have suffixes on their device files that designate their particular drive type. For example:

`/dev/rmt/tps`*Controller*`d`*Target*`nrv.8500c` (EXB8500C)

Examples of No Rewind Device Files

Example 1

If the desired HP 4-mm (DAT) drive is on SCSI bus 1 at SCSI ID 4, you specify the following device path for that drive:

```
/dev/rmt/tps1d4nrv
```

Example 2

If the desired Exabyte 8500C or 8505 tape drive is on SCSI bus 0 at SCSI ID 3, you specify the following device path for that drive:

```
/dev/rmt/tps0d3nrv.8500c
```

Example 3

If the desired DLT2000 or DLT4000 tape drive is on SCSI bus 0 at SCSI ID 5, you specify the following device path for the drive:

```
/dev/rmt/tps0d5nrvc
```

Example 4

If the desired DLT7000 tape drive is on SCSI bus 0 at SCSI ID 5, you specify the following device path:

```
/dev/rmt/tps0d5nrvc.7000c
```

Example 5

If the desired Exabyte 8900 (Mammoth) is on SCSI bus 1 at SCSI ID 5, you specify the following device file path for the drive:

```
/dev/rmt/tps1d5nrvc
```

Since this drive writes in only one format, you can ignore other device files that are created for this drive.

Adding HP 4-mm Drives and HP C1560B DAT Autoloaders

Read this topic if you plan to use standalone or robotic Hewlett-Packard (HP) 4-mm DAT tape drives or HP C1560B DAT Autoloaders. It explains drive switch settings and kernel changes you may have to make in order for the system to recognize these devices.

Checking Switch Settings

Ensure that the hardware (tape drive) switch settings on HP35480A 4-mm (DAT) drives are as follows:

Note: Other combinations may work, but these are the settings that were functional during testing by VERITAS with an HP35480A drive and HP C1560B Autoloader.

On=1, Off=0

Switch	Setting
1	1
2	1
3	1
4	1
5	1
6	1
7	0
8	0

Ensure that the hardware (tape drive) switch settings on the HP C1533A 4-mm (DAT) drives are as follows:

Switch	Setting
1	1
2	1
3	0
4	1
5	1
6	1
7	0
8	0

Changing `/var/sysgen/master.d/scsi`

For the system to recognize the 4-mm DAT drives, the `struct tpsc_types tpsc_types[]` array must have code entries for them. You will find this array in the `/var/sysgen/master.d/scsi` file.

1. The code entries that must be in this array are as follows:

   For all DAT drives except an HP C1560B DAT Autoloader:

```
/* HP DAT drives. Any product number that starts with HP354.*/
{ DATTAPE, TPDAT, 2, 5, "HP", "HP354", 0, 0, {0, 0, 0, 0},
MTCAN_BSF|MTCAN_BSR|MTCAN_APPEND|MTCAN_SETMK|MTCAN_PART|MTCAN_PREV|
MTCAN_SYNC|MTCAN_SPEOD|MTCAN_CHKRDY|MTCAN_VAR|MTCAN_SETSZ|
MTCAN_SILI|MTCAN_SEEK|MTCAN_CHTYPEANY,
/* minimum delay on i/o is 4 minutes, because when a retry is
* performed, the drive retries a number of times, and then
* rewinds to BOT, repositions, and tries again.  */
40, 4*60, 4*60, 5*60, 512, 128*512, 0, (u_char*)0, 3 * 3600,
(0), 0, 0, 0,
},
```

   For an HP C1560B DAT Autoloader:

```
/* HP DAT drives. Any product number that starts with HP1533. */
 { DATTAPE, TPDAT, 2, 5, "HP", "C1533", 0, 0, {0, 0, 0, 0},
 MTCAN_BSF|MTCAN_BSR|MTCAN_APPEND|MTCAN_SETMK|MTCAN_PART|MTCAN_PREV|
 MTCAN_SYNC|MTCAN_SPEOD|MTCAN_CHKRDY|MTCAN_VAR|MTCAN_SETSZ|
 MTCAN_SILI|MTCAN_SEEK|MTCAN_CHTYPEANY,
 /* minimum delay on i/o is 4 minutes, because when a retry is
 * performed, the drive retries a number of times, and then
 * rewinds to BOT, repositions, and tries again.  */
 40, 4*60, 4*60, 5*60, 512, 128*512, 0, (u_char*)0, 3 * 3600,
 (0), 0, 0, 0,
 },
```

2. If the above code is in `/var/sysgen/master.d/scsi` and you have previously rebuilt the kernel as explained in Step c of Step 3 below, then no further changes are necessary.

3. If the code is not in `/var/sysgen/master.d/scsi`, add it as follows:

   a. Save a copy of `/var/sysgen/master.d/scsi`.

   b. Add the above code. The easiest way to make this addition is to copy it from the `dev_config_guide.txt` file.

   c. After completing your changes to the `/var/sysgen/master.d/scsi` file, reconfigure the kernel by running the kernel auto-configuration script:

      ```
      /etc/autoconfig
      ```

   d. Reboot the system to utilize the newly built kernel.

Fast-Tape Positioning (locate-block)

For most drive types, Media Manager supports the SCSI `locate-block` command for positioning a tape to a specific block. This improves tape-positioning times over the alternative method, which is the forward-space-file / record method. See the *NetBackup Release Notes - UNIX* for a list of drive types that support `locate-block`.

NetBackup and Storage Migrator use the `locate-block` command by default unless you disable the command by executing the following:

```
touch /usr/openv/volmgr/database/NO_LOCATEBLOCK
```

With locate-block positioning disabled, NetBackup uses the forward-space-file / record method and Storage Migrator skips file marks.

Adding Sony DTF Drives

For the system to recognize the DTF drives, the code in the struct `tpsc_types tpsc_types[]` array must contain entries for them. You will find this array in the `/var/sysgen/master.d/scsi` file.

1. The code entry that must be in this array follows:

```
/*  Sony DTF drive */
{ DECDLT, TPDLT, 4, 7, "SONY", "GY-2120", 0, 0,
(0), MTCAN_BSF|MTCAN_BSR|MTCAN_APPEND|MTCAN_SPEOD|
MTCAN_CHKRDY|MTCAN_VAR|MTCAN_SETSZ|MTCAN_SILI|MTCAN_SEEK|
MTCAN_SYNC|MTCAN_CHTYPEANY|MTCAN_COMPRESS,
20, 8*60, 20*60, 5*60, 4096, 64*1024, 0, (u_char *)0, 3 * 3600,
(0), 0, 0, 0,
},
```

2. If the above code is in `/var/sysgen/master.d/scsi` and you have previously rebuilt the kernel as explained in Step c of Step 3 below, then no further changes are necessary.

3. If the code is not in `/var/sysgen/master.d/scsi`, add it as follows:

   a. Save a copy of the `/var/sysgen/master.d/scsi` file.

   b. Add the above code. The easiest way to make this addition is to copy it from the `dev_config_guide.txt` file.

   c. After completing your changes to the `/var/sysgen/master.d/scsi` file, reconfigure the kernel by running the following kernel auto-configuration script:

      `/etc/autoconfig`

   d. Reboot the system to utilize the newly built kernel.

Configuring Optical Disk Drives

When adding optical disk drives to a Media Manager configuration, you must specify the following device paths:

- Character device path (disk partition `s7`)
- Volume header disk device path (disk partition `vh`)

In a typical SGI IRIX configuration, most of the desired optical disk device files already exist and you just have to locate them in the `/dev` directory.

Character disk device files have the following format:

`/dev/rdsk/dks`*Controller*`d`*Target*`s7`

Volume disk device files have the following format:

`/dev/rdsk/dks`*Controller*`d`*Target*`vh`

Where:

Controller is the SCSI bus (adapter) number.

Target is the SCSI ID.

`s7` is the desired character device partition.

`vh` is the desired volume header partition.

Examples of Optical Disk Device Files

If the desired optical disk drive is on SCSI bus 1 at SCSI ID 3, you specify the following paths:

`/dev/rdsk/dks1d3vh` (volume header)
`/dev/rdsk/dks1d3s7` (character device)

Command Summary

The following is a summary of commands that may be useful when configuring devices. See the procedures in this chapter for examples of their usage.

`/usr/etc/mediad -k`

`/usr/etc/mediad`
Edit and modify `/etc/config/mediad.config`.
This command does not take effect until the next time you start `mediad`. Use the `-k` option to stop `mediad`. Restart `mediad` by executing `mediad` with no options.

`MAKEDEV` *type*
If the device files you need do not exist, you can execute the `MAKEDEV` command from the `/dev` directory to create them.
type indicates the type of device file, as follows:
`tps` creates all the tape device file combinations for `tps` (the SCSI tape driver for Integral SCSI controllers)
`scsi` creates all the device files for the generic SCSI driver.
`dks` creates all the device files for dks (the SCSI disk driver for integral SCSI controllers).

`/etc/autoconfig`
Runs the kernel auto-configuration script.

`/usr/openv/volmgr/bin/vmconf`
Provided with Media Manager, this script eases device setup in less complex configurations.

`/sbin/hinv`
Shows the system configuration, including devices configured on SCSI controllers.

DEC Alpha Running Digital UNIX 4.0878

This chapter explains how to configure devices for use with Media Manager on a DEC Alpha platform running Digital UNIX. You configure drives and robots using one of the available Media Manager administrative interfaces.

The main topics included in this chapter are:

- Configuring Robotic Controls
- Configuring Tape Drives
- Adding Nonstandard Tape Drives
- Command Summary

To determine which drive and robot types are supported, refer to the *NetBackup Release Notes - UNIX*.

Note: VERITAS does not recommend or support the use of single-ended to differential SCSI converters on Media Manager controlled devices. You may encounter problems if you use these converters.

You can copy code examples from the on-line version of this manual, located in `/usr/openv/volmgr/dev_config_guide.txt`. This file is installed with Media Manager software.

Configuring Robotic Controls

Robots are controlled through a SCSI, an RS-232, or a network connection. Configuration for network controlled robotic libraries is discussed in the appendices of the *Media Manager System Administrator's Guide - UNIX*. SCSI and RS-232 control is covered in the following sections.

Configuring SCSI Robotic Controls

Read this topic if you plan to use a robotic storage device that is controlled through a SCSI robotic connection. See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with these supported SCSI robot types.

- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TL4 - Tape Library 4MM
- TL8 - Tape Library 8MM
- TLD - Tape Library DLT
- TS8 - Tape Stacker 8MM
- TSD - Tape Stacker DLT

When communicating with SCSI-controlled robotic peripherals, Media Manager robotic software utilizes the generic (user mode) SCSI pass-through driver. The Digital UNIX kernel does not have to be reconfigured to use this driver, since this driver is part of basic Digital UNIX.

Media Manager does require that a special file be created in the `/dev` directory for SCSI controlled robotics. If the `/usr/openv/volmgr/bin/vmconf` script is used to configure devices, it creates the necessary device files.

If you do not use this script, the device files must be created manually using the `mknod` command as follows:

```
cd /dev
/sbin/mknod robtypecbusttargetllun   c 38 minor
```

Where:

robtype is the robot type in lower case (for example, tsd).

bus is the bus (adapter) number.

target is the SCSI ID

lun is the logical unit number (*lun* is always 0, except for DLT2700, DLT4700, HP C1560B, and some other peripherals).

minor equals (`bus * 256) + (target * 16) + lun`

Examples of SCSI Robotic Control Device Files

Example 1

If the robotics control for an Exabyte 10i (TS8) is connected to bus 0 at SCSI ID 5, lun 0, the commands to create the device file are as follows:

```
cd /dev
/sbin/mknod ts8c0t5l0 c 38 80
```

This creates the following device file, which you specify to Media Manager:

```
/dev/ts8c0t5l0
```

Example 2

If the robotics control for a Quantum DLT2700 (TSD) is connected on bus 1 at SCSI ID 3, lun 1, the commands to create the device file would be:

```
cd /dev
/sbin/mknod tsdc1t3l1 c 38 305
```

This creates the following device file, which you specify:

```
/dev/tsdc1t3l1
```

The `lsdev` command in `/usr/openv/volmgr/bin` can be used to determine what devices are physically connected to the system. An example for determining connected autochangers follows:

```
/usr/openv/volmgr/bin/lsdev changer
```

This example shows that there is only one possible autochanger connected to this system.

```
Bus 0 Scsi Id 5 Lun 0, Changer: EXABYTE EXB-10i 3.0
```

Configuring RS-232 Robotic Controls

Read this topic if you plan to use a robotic device that is controlled through an RS-232 connection. See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with the following supported RS-232 robot types:

- TC4 - Tape Carousel 4MM
- TC8 - Tape Carousel 8MM
- TLD - Tape Library DLT

Configuring RS-232 controlled devices on the DEC Alpha platform normally consists of choosing between the COM1 or COM2 serial ports.

Example of RS-232 Robotic Control Device Files

Typical path names to be used when adding a RS-232 controlled robotic device to the configuration are as follows:

```
serial port 1: /dev/tty00
serial port 2: /dev/tty01
```

The device files for the serial ports should exist once the system is installed. If they do not, they can be created with the following commands:

```
cd /dev
./MAKEDEV ace0
```

Configuring Tape Drives

Fast-Tape Positioning (locate-block)

For most drive types, Media Manager supports the SCSI `locate-block` command for positioning a tape to a specific block. This improves tape-positioning times over the alternative method, which is the forward-space-file / record method. See the *NetBackup Release Notes - UNIX* for a list of drive types that support `locate-block`.

NetBackup uses the `locate-block` command by default unless you disable it by executing the following:

```
touch /usr/openv/volmgr/database/NO_LOCATEBLOCK
```

With locate-block positioning disabled, NetBackup uses the forward-space-file / record method.

Adding Standard Tape Drives

When adding tape drives to a Media Manager configuration, you need only specify a no rewind on close device path. These device files are located in the `/dev` directory, and have the following format:

`/dev/nrmt`*LtuDensity*

Where:

Ltu is the logical tape unit. When the first `MAKEDEV` of a tape drive is done, *Ltu* is 0. The next time, *Ltu* is 1 and so on.

Values for *Density* can be `l`, `m`, `h`, or `a`. Typically, `h` (high) is used.

If the desired tape device file does not exist, device files can be created using the `MAKEDEV` command as follows:

```
cd /dev
./MAKEDEV tz  (bus * 8) + SCSI ID
```

Media Manager provides a command that you can use to determine the devices that are physically connected to the system. This command is located in `/usr/openv/volmgr/bin` and is named `lsdev`.

An example of using `lsdev` to determine connected tape drives follows:

```
lsdev tape

Bus 0 Scsi Id 3 Lun 0, Tape (rmt2): EXABYTE EXB-8500-85Qanx005E0
Bus 0 Scsi Id 4 Lun 0, Tape (rmt0): EXABYTE EXB-850085QANXRC05E0
```

Alternatively, you can also use the following form of the command:

```
lsdev logical_tape_devs

rmt2 is defined on bus 0, scsi id 3
rmt0 is defined on bus 0, scsi id 4
```

If the device files do not exist for a connected tape drive, the command shows (----) instead of rmt*Ltu*, for example:

```
lsdev tape

Bus 0 Scsi Id 3 Lun 0, Tape (rmt2): EXABYTE EXB-8500-85Qanx005E0
Bus 0 Scsi Id 4 Lun 0, Tape (----): EXABYTE EXB-850085QANXRC05E0
```

In the above example, the device files for the tape drive on bus 0, SCSI ID 4 do not exist. To create device files, you use the MAKEDEV command.

```
cd /dev
./MAKEDEV tz4

MAKEDEV: special file(s) for tz4:
rmt0l
rmt0h
rmt0m
rmt0a
nrmt0l
nrmt0h
nrmt0m
nrmt0a
```

Examples of No Rewind Device Files

Example 1

If the desired Exabyte 8500 tape drive is on bus 0 at SCSI ID 4, the commands to create the device files follow:

```
cd /dev
./MAKEDEV tz4
```

This creates the following device file, which you specify to Media Manager (this example assumes *Ltu* is 0):

```
/dev/nrmt0h
```

Example 2

If the desired DLT4000 tape drive is on bus 1 at SCSI ID 3, the commands to create the device files are as follows:

```
cd /dev
./MAKEDEV tz11
```

This creates the following device file, which you specify to Media Manager (this example assumes *Ltu* is 1):

```
/dev/nrmt1h
```

Adding Nonstandard Tape Drives

VERITAS has tested several tape drives on DEC Alpha Digital UNIX, including EXABYTE 8-mm drives, HP 4-mm DAT drives, and Quantum DLT drives. Normally, using tape drives from these vendors does not require kernel reconfiguration because the default definitions are sufficient.

If a drive vendor recommends kernel reconfiguration, the file that contains the tape drive definitions is `/usr/sys/data/cam_data.c`. If this file is modified, care should be taken to ensure tape drives are configured in variable (rather than fixed) mode.

If the `/usr/sys/data/cam_data.c` file has been changed, refer to the `doconfig(8)` command for information on rebuilding a new kernel.

Switch Settings for HP C1533A 4mm DAT Drives

If you have standalone or robotic 4MM drives that are model HP C1533A, you may have to change the switch settings on the bottom of the drive. This is the drive that comes in the HP C1560B (48AL) DAT Autoloader.

If the drive or autoloader was purchased from Hewlett Packard, the default switch settings should work. However, if the drive or autoloader were purchased from some other vendor, that vendor may have changed the default switch settings. The same thing may apply to other vendor's 4MM robots if they contain HP C1533A drives.

If this situation exits, set the switch settings to the following (this is the documented default):

On=1, Off=0

Switch	Setting
1	1
2	1
3	0
4	1
5	1
6	1
7	1
8	1

NetBackup 3.2 Media Manager Device Configuration Guide for UNIX

Command Summary

The following is a summary of commands that may be useful when configuring devices. See the procedures in this chapter for examples of their usage.

`/sbin/mknod` *robtype*`c`*bus*`t`*target*`l`*lun* `c 38` *minor*

Execute this command from the `/dev` directory to create the special device file for SCSI controlled robotics. If the `/usr/openv/volmgr/bin/vmconf` script is used to configure devices, it automatically creates the necessary device files and this command is unnecessary.
Where:
robtype is the robot type in lower case (for example, ts8).
bus is the bus (adapter) number.
target is the SCSI ID.
lun is the logical unit number (*lun* is 0, except for DLT2700, DLT4700, HP C1560B, and some other peripherals).
minor equals `(bus * 256) + (target * 16) + lun`

`MAKEDEV ace0`

Creates device files for the serial ports. Normally, these files exist after the system is installed. Execute this command from the `/dev` directory.

`MAKEDEV tz` *(bus * 8) + SCSI ID*

Creates device files for tape drives. Execute this command from the `/dev` directory.

`/usr/openv/volmgr/bin/lsdev tape`

Displays tape devices that are physically connected to the system.

`/usr/openv/volmgr/bin/vmconf`

Provided with Media Manager, this script eases device setup in less complex configurations.

NCR Running MP-RAS 3.02

9

This chapter explains how to configure devices for use with Media Manager on a NCR system. You configure drives and robots using one of the available Media Manager administrative interfaces.

The main topics covered here are as follows:

- NCR Device Files
- Configuring Robotic Controls
- Configuring Tape Drives

NCR Device Files

You do not need to install a pass-through driver or run `mknod` commands to add new device files. (The device files are created automatically when the machine is rebooted after adding a new device.)

After you attach the hardware and boot the machine, locate your device file names in `/etc/device.tab.rd` and use those device file names when configuring Media Manager.

Information about attached devices can be found in the text file `/etc/device.tab.rd` as follows:

```
------snippet 1 from /etc/device.tab.rd ------
c13t2d0s0:/dev/rmt/c13t2d0s0:::\
        removable="true" \
        id="Quantum  DLT4000         " \
        desc="Tape Drive" \
-------------------------------------------

------snippet 2 from /etc/device.tab.rd ------
c13t4d0s0:/dev/rchg/c13t4d0s0:::\
        removable="true" \
        id="STK      9714            " \
        desc="Medium Changer Device" \
-------------------------------------------
```

Configuring Robotic Controls

Robots are controlled through a SCSI or a network connection. Configuration for network controlled robotic libraries is discussed in the appendices of the *Media Manager System Administrator's Guide - UNIX*.

From the previous example, an example robotic path for SCSI control is `/dev/rchg/c13t4d0s0`.

Configuring Tape Drives

To configure a no rewind on close tape device, use the device file with the `nn` suffix. In the following example this device file would be: `/dev/rmt/c13t2d0s0nn`.

The following example list was created using `/usr/openv/volmgr/bin/tpconfig -d`:

```
Index  DriveName           DrivePath             Type   Multihost    Status
*****  **********          *********             ****   *********    ******

  0    DRIVE2              /dev/rmt/c13t2d0s0nn  dlt      No          UP
       TLD(0) Definition     DRIVE=2

Currently defined robotics are:
  TLD(0)     robotic path = /dev/rchg/c13t4d0s0, volume database host = ted
```

Note: The list of currently supported devices is limited. The list includes: STK9710 and STK9714 robots (SCSI or Automated Cartridge System control) with DLT2000/DLT4000 drives. More devices may be added later.

NetBackup 3.2 Media Manager Device Configuration Guide for UNIX

Sequent Running DYNIX/ptx 4.2.1/4.2.3/4.4.2/4.4.4

This chapter explains how to configure devices for use with Media Manager on a Sequent system running DYNIX. You configure drives and robots using one of the available Media Manager administrative interfaces.The main topics covered here are as follows:

- Configuring Robotic Controls
- Configuring Tape Drives

To determine which drives are supported on a Sequent system, refer to the *NetBackup Release Notes - UNIX*.

Configuring Robotic Controls

Robots can be controlled through a SCSI or a network connection. Configuration for network controlled robotic libraries is discussed in the appendixes of the *Media Manager System Administrator's Guide - UNIX*. These appendixes describe specific platform requirements and restrictions.

SCSI robotic control is covered in the following section.

Configuring SCSI Robotic Controls

The following SCSI robots are supported on Sequent levels 4.4.2 or 4.4.4 only. See the *NetBackup Release Notes - UNIX* for a list of the vendor models associated with these robot types.

- TL4
- TL8
- TLD
- TS8

Use the following procedure to configure a pseudo device file for the robot pass-through capability:

1. The following display using `lsdev,` lists the devices in a system. `lsdev` uses the pass-through capability to do an inquiry command. If this command works it is a good sign that the robotics will also work.

```
/usr/openv/volmgr/bin/lsdev
Bus 0, target 0, lun 0, Disk: (IBM OEM DFHSS4E         4343)
Bus 0, target 1, lun 0, Disk: (SEAGATE ST15150W        0023)
Bus 0, target 3, lun 0, Tape: (EXABYTE EXB8500C8SQANXRU07J0)
Bus 0, target 4, lun 0, Tape: (TANDBERG TDC 3800       -07:)
Bus 0, target 5, lun 0, Cdrom: (PLEXTOR CD-ROM PX-6XCS  4.05)
Bus 0, target 7, lun 0, Processor: (SEQUENT CSM SCSI Ctlr 0601)
Bus 0, target 8, lun 0, Disk: (HP     C2490A          5083)
Bus 1, target 1, lun 0, Disk: (SEAGATE ST15150W        0023)
Bus 1, target 3, lun 0, Changer: (STK     9730          1102)
Bus 1, target 4, lun 0, Tape: (Quantum DLT4000         CD3C)
Bus 1, target 5, lun 0, Tape: (Quantum DLT4000         CD3C)
```

2. Note the bus, target, and lun of the robotic library you want to control as a TLD robot. In the above example, it is the STK 9730.

3. Create a pseudo device file, as follows:

   a. Create a directory in `/dev`.

      ```
      cd /dev
      ```

      ```
      mkdir dir-name
      ```

      ```
      cd dir-name
      ```

   b. Create a file, *file-name*, in this directory that contains the bus, target, and lun for the robotics.

      The directory name and file name used in the following example is `veritas/stk9730`, but they can be any names. To configure the STK 9730 robot, create a file as follows. The `lsdev` display above shows that the bus is 1, the target is 3, and the lun is 0. These three values are entered in the new file.

      ```
      cat > stk9730
      ```

      ```
      1 3 0
      ```

      ```
      ^D
      ```

4. Use `/dev/`*dir-name*`/`*file-name* as the robotic path when using `tldtest` or when configuring the robot. For example:

   ```
   tldtest -r /dev/veritas/stk9730
   ```

   Media Manger reads the file to obtain the path to the device required by the pass-through capability (bus, target, and lun).

Configuring Tape Drives

The `vmconf` configuration script does not support adding tape drives or robots to the Media Manager configuration on Sequent systems.

The following tape devices are supported:

Table 1 Supported Tape Devices

Drive Type	Sequent Driver
Exabyte 8500	tx
Exabyte 8500C	tx
Exabyte 8505	tx
Exabyte 8505XL	tx
Exabyte 8900	tx
DLT4000	tl
DLT7000	tl
IBM Magstar (3590)	tc
4mm DAT	td

See the Sequent DYNIX `man` pages on the tape drivers for information on which device paths to use for a specific drive.

Table 2 Example Device Files for Media Manager

Drive Type	No Rewind Device
Exabyte 8500C	/dev/rmt/tx0x85cn
1/2 Cartridge (3480)	/dev/rmt/tf2n
QIC 150	/dev/rmt/tm0q150n
DLT	/dev/rmt/tl4n
IBM Magstar (3590)	/dev/rmt/tc3n
4mm DAT	/dev/rmt/td6n

Kernel Configuration

Media Manager (`avrd`) periodically attempts to open configured tape drives that are UP, to see if a tape has been loaded. DYNIX logs error messages to the console when a not ready (empty) tape drive is opened.

Described below are kernel configuration options you can make to reduce the number of messages that are logged.

After making changes to the kernel configuration file you must generate a new kernel for the system. See the config(1M) man page.

Turning Off Messages

To turn off messages for drives being scanned, change the following line in `/usr/conf/uts/io/scsitape/scsitape_space.c`:

From

```
int sct_devroute = CE_TRACE | CE_WARN;
```

to

```
int sct_devroute = CE_TRACE;
```

Exabyte Drive Type

If you are using 8mm Exabyte tape drives you may want to disable the 45 second wait for a drive to become ready. You need to set the configuration value `tx_ready_timeout` to 0 in the `/usr/conf/uts/io/tx/tx_space.c` kernel configuration file as follows:

Change

```
int tx_ready_timeout = 45;
```

to

```
int tx_ready_timeout = 0;
```

DLT Drive Type

If you are using DLT tape drives you may want to disable the 45 second wait for a drive to become ready. You need to set the configuration value `tl_ready_timeout` to 0 in the `/usr/conf/uts/io/tl/tl_space.c` kernel configuration file.

Change

```
int tl_ready_timeout = 45;
```

to

```
int tl_ready_timeout = 0;
```

Tape Drive Support

DLT Drive Type

The DLT driver from Sequent should be installed. Refer to the Sequent Computer Systems installation guide for installation instructions for this driver.

IBM Magstar (3590) Drive Type

The IBM Magstar driver from Sequent should be installed. Refer to the Sequent Computer Systems installation guide for installation instructions for this driver.

Command Summary

The following commands display the hardware configuration.

`/etc/dumpconf`

Examines the physical devices configured on the system.
The `-d` option shows the SCSI buses and tape devices on the system.

`/etc/showcfg`

Displays the configuration of the system in a manner similar to the power-up monitor configuration command.
The `-s` option selects an alternate one-line format that gives the quantity of each type of board.
The `-d` option produces a dump of relevant parts of the system configuration description table. The data displayed includes information about the memory available, the boot flags, the boot device, console tty control characters, and the current system bus mode.

Pyramid RM1000 Running Reliant UNIX 5.43 B0078

This chapter explains how to configure devices for use with Media Manager on a Pyramid RM1000 running Reliant UNIX. You configure drives and robots using one of the available Media Manager administrative interfaces.

The main topics included in this chapter are:

- Configuring Robotic Controls
- Configuring Tape Drives

Configuring Robotic Controls

No robots are supported with direct control. Configuration information for network-controlled robots can be found in the Automated Cartridge System appendix and in the TLM (Grau) appendix in the *Media Manager System Administrator's Guide - UNIX*.

Configuring Tape Drives

When adding tape drives to a Media Manager configuration, you need only specify a no rewind on close device path. These device files are located in the `/dev/tape` directory and have the following format:

```
/dev/tape/rt2cn
```

Using Berkeley-style Close

You must specify a Berkeley-style close for tape devices that you configure under Media Manager.

The terms *Berkeley-style close* and *AT&T close* refer to where a tape is left logically positioned after a close operation (in relation to a tape mark). One style leaves an application logically positioned before a tape mark and the other leaves it after. Applications must assume where the tape is left after a close in order to establish the correct orientation the next time they do a tape position or read operation. Some operating systems allow tape devices to be configured with either type of close. NetBackup assumes it is using a Berkeley-style close on a Pyramid RM1000.

Checking For Berkeley-style Close

To determine if a tape device is set to Berkeley-style close, follow these steps:

1. Use the following command to list the available tape devices:

   `autoconf -l -n` *node_name*

   Where:

   `-l` means list all devices

node_name is the name of the RM1000 cell node running Media Manager software.

The output will be similar to the following:

```
System Configuration:
   NCR 720 SCSI-2 Controller Rev:x.xx Path:0 Id:7
xpt0
t1 Pyramid Model 3466 (8mm tape)
    NCR 720 SCSI-2 Controller Rev:x.xx Path:1 Id:6
    xpt1
t2 Pyramid Model 3445 (128-trk wide tape drive)
```

This node has two tape devices configured, t1 (8mm tape) and t2 (128-trk wide tape drive).

2. Use the following command to list the characteristics of the required tape device:

```
vtconfig -p t2
```

The output will be similar to the following:

```
/*
 Physical tape configuration for
 */
physical tape{
   filemark = default_file_tapemark; mts_type = 5; erase_op =
   remaining_partition_logical;
   block_mode = variable;
   gap_size = default_size;
   eod = one_consecutive_tapemark generate_with_filemarks;
       node   name_suffix = "d0c"          <== Rewind Device
              density = 0x0
              compression = 0x1
              close_action = rewind
              mts_density = low
              alias_suffix = "c";

       node   name_suffix = "d0cn" <== No rewind Device
              density = 0x0
              compression = 0x1
              close_action = none         <== Close Action
              mts_density = low
              alias_suffix = "cn";

}
```

For a Berkeley-style close, the close action for the no rewind on close device must be set to none.

Setting Berkeley-style Close

If the close action is set to forward_space_filemark, an AT&T style close is used. In this case, the device must be reconfigured to use Berkeley-style close as follows:

1. Use the following command to remove the device file from `/dev/tape`:

   `tpadmin -d t2`

2. Rename the tape device:

   `tpadmin -n`

3. Configure the device as Berkeley-style close:

   `vtconfig -c` *tape_device* `PTC_128trkC_BSD /dev/phys/tape/`*tape_device*

   Where *tape_device* is the tape device name. For example, t2.

   This command will use the file `PTC_128trkC_BSD` in `/etc/default/tapeinfo/vtconfig` to define the close action. This command also recreates `/dev/tape/rt2c` and `/dev/tape/rt2cn`.

4. Check the close action:

   `vtconfig -p` *tape_device* `| more`

   Where *tape_device* is the tape device name.

See the `vtconfig`, `tpadmin`, and `autoconf` man pages for further information.

Glossary

access control list (ACL)
security information associated with files on some file systems.

ACS
Automated Cartridge System.

active job
A job for which NetBackup is currently processing backup or restore data.

activity logs
Logs that can be optionally enabled for specific NetBackup programs and processes and then used to investigate problems.

activity monitor
A NetBackup administration utility on Windows NT that shows information about NetBackup jobs and provides limited control over them.

administrator
A user that is granted special privileges to install, configure, and manage the operation of a system, network, or application

administration client
A Windows NT NetBackup client that has the administration interface software installed and can be used to administer NetBackup servers.

alternate-client restore
Restores files to a different client than the one from which they were backed up.

alternate-target restore
On a Novell NetWare server platform running the NetBackup target version of client software, this operation restores files to a different target than the one from which they were backed up.

alternate path restore
Restores files to a different directory than the one from which they were backed up.

archive
A special kind of backup where NetBackup backs up the selected files, and if the backup is successful, deletes the files from the local disk. In this manual, references to backups also apply to the backup portion of archive operations except where otherwise noted.

archive bit
A file-status bit that the Microsoft based operating system sets when it writes a file, thereby indicating that the file has changed.

attributes for a class
Configuration parameters that control the behavior of NetBackup during operations involving this class.

automatic backup
A scheduled backup by the master server.

backup
Refers to the process of copying and saving files and directories to storage media. For example, "the backup is complete" This term can also refer to the collection of data that NetBackup saves for a client during a backup or archive. For example, "duplicate the backup" (also see image).
Backup is two words when used as a verb. For example, "back up the file."

backup, archive, and restore interface
The name of the NetBackup Microsoft Windows based and Java based user interfaces for clients. On servers, these interfaces can be started through the NetBackup Administration interface.

backup window
The period of time during which backups can begin.

block size
The number of bytes in each block of data written on the media during a backup.

bp
A backup, archive, and restore utility for users on NetBackup UNIX clients. It has a character-based, menu interface that can be run from terminals that do not have X Windows capabilities.

bpadm
An administrator utility that runs on NetBackup UNIX servers. It has a character-based, menu interface that can be run from terminals that do not have X Windows capabilities.

bp.conf file
A NetBackup configuration file on UNIX servers and also on UNIX, Macintosh, and OS/2 clients.

bpcd
NetBackup Client service on Windows NT and the NetBackup Client daemon on UNIX.

bprd
NetBackup Request Manager service on Windows NT and NetBackup Request daemon on UNIX.

catalogs
Internal NetBackup or Media Manager databases on a Windows NT NetBackup server. These databases contain information about configuration, status, errors, and the files and directories in the stored backup images.

CDF
Context-dependent file, which is a type of directory structure on a Hewlett-Packard system.

class
Defines the backup policy for a group of one or more clients that have similar backup requirements.

client
The system with the files to back up, archive, or restore.

client-user interface
The program used to perform user backups, archives, and restores.

command lines
Commands that users can execute either from the command line or in scripts.

compression
The process of compacting data to enable more efficient transmission and storage.

cpio
A UNIX command for formatting data on a tape.

ctime
The time that a UNIX inode was changed.

cumulative-incremental backup
A backup that is scheduled by the administrator on the master server and backs up all files that have changed since the last successful full backup. All files are backed up if no prior backup has been done.

daemon
A program on a UNIX system that runs in the background and performs some task (for example, starting other programs when they are needed). Daemons are generally referred to as services or processes on Windows NT NetBackup servers and clients.

database-extension clients
Clients with additional NetBackup software that is designed to back up relational databases.

databases for NetBackup or Media Manager
Internal NetBackup or Media Manager databases. The NetBackup databases contain information about configuration, status, errors, and the files and directories in the stored backup images. The Media Manager databases contain configuration information on the robots, devices and volumes that are under control of Media Manager. .

debug logs
See “activity logs."

device delays
Delays caused by the device that are beyond the control of the storage application. An example is the time required to position tape under the read and write heads.

device monitor
A Media Manager administration utility that provides manual control of Media Manager storage devices. For example, an administrator or computer room operator can use this utility to manually reset devices or set them to the UP or DOWN state.

DHCP
Dynamic host configuration protocol. This TCP/IP protocol automatically assigns temporary IP addresses to hosts when they connect to the network.

differential-incremental backup
Scheduled by the administrator on the master server and backs up files that have changed since the last successful full or incremental backup. All files are backed up if no prior backup has been done.

directory tree
The hierarchical structure in which files are organized on a disk. Each directory lists the files and directories that are directly below it in the tree. On UNIX, the topmost directory is called the root directory.

disaster recovery
Recovering data from backups after a disk crash or other catastrophe.

disk-image backup
A bit-by-bit rather than a file system backup of a disk drive on Windows NT.

DLT
Digital-linear tape or tape drive.

Domain Name Service (DNS)
A program that handles name translation for network communications.

drive cleaning
The use of a special cleaning tape to clean the heads on a drive.

duplicate image
A copy of a backup image.

encryption
Provides additional security by encrypting backup data on the client. This capability is available only with the NetBackup Encryption option.

exclude list
A list that designates files or directories to exclude from automatic backups.

expiration (image)
The date and time when NetBackup stops tracking a backup image.

EVSN
External volume serial number. This is an identifier written on a media cartridge or canister so the operator can identify the volume before inserting it into a drive or robot. For labeled media, the EVSN must be the same as the RVSN (identifier recorded on the media).

FastBackup
A special type of raw-partition backup that can be performed only on an Auspex client.

FlashBackup
A special type of raw-partition backup that is available only with the NetBackup FlashBackup option.

fragment
A part of a backup or archive image. NetBackup can be configured to divide images into fragments when they exceed a certain size or span tapes.

frequency (backup)
How often NetBackup performs scheduled backups. For example, if the frequency is seven days then backups occur once a week.

FROZEN media state
If a volume is FROZEN, NetBackup keeps it indefinitely and can restore from it but not use it for further backups or archives.

full backup
Copies to a storage unit all files and directories that are beneath a specified directory.

FULL media state
If a volume is FULL, it cannot hold more data or be used for further backups.

global attributes
NetBackup configuration attributes that affect all classes.

Global Data Manager
A separately priced option that provides an interface with a tree view where you can select master servers to administer. The server where the option is installed is called a Master of Masters.

GNU tar
A public domain version of the UNIX tar program.

goodies directory
A directory containing programs, scripts, and other files that are not formally supported.

gravity stacker
A robot that relies on gravity to advance to the next required tape.

GUI
Graphical user interface.

hard link
On UNIX, a hard link is a pointer to the inode for the data.

hierarchical storage management
The process of automatically migrating selected files from a managed file system to specified migration levels on secondary storage, while maintaining transparent access to those files.

host name
Name by which a host computer is identified by programs and other computers in the network.

HSM
See storage migrator.

image
The collection of data that NetBackup saves for an individual client during each backup or archive. The image contains all the files, directories, and catalog information associated with the backup or archive.

import
The process of recreating NetBackup records of images so the images can be restored.

include list
A list that designates files or directories to add back in from the exclude list.

incremental backup
See “cumulative-incremental backup" and “differential-incremental backup."

inode
A UNIX data structure that defines the existence of a single file.

install_path
Directory where NetBackup and Media Manager software is installed. The default on Windows NT is `C:\VERITAS` and on UNIX it is `/usr/openv`.

jbp
A Java-based interface for performing user backups, archives, and restores.

Job
A parcel of work submitted to a computer. NetBackup jobs are backups, archives, or restores.

kernel
The nucleus of an operating system.

keyword phrase
A keyword phrase is a textual description of a backup.

kill a job
Terminating a job and removing it from the job queue.

label
Identifies a volume and includes a media ID.

library
Refers to a robot and its accompanying software. A library includes a collection of tapes or optical platters used for data storage and retrieval. For example, a Tape Library DLT (TLD) refers to a robot that has TLD robotic control.

link
See “hard link" or “symbolic link."

LMF - Library Management Facility
A Media Manager designation for a category of robot. For the specific vendor types and models in this category, see the NetBackup release notes.
(noun) Amount of work that is being performed by a system or the level of traffic on a network. For example, network load affects performance.

(verb) **logs**
Files where a computer or application records information about its activities.

man pages
Online documentation provided with UNIX computer systems and applications.

Master of Masters
A NetBackup server that has an interface with a tree view where you can select master servers to administer. The Master of Masters requires the Global Data Manager option.

master server
NetBackup server that provides administration and control for backups and restores of client data.

media
Physical magnetic tapes, optical disks, or magnetic disks where data are stored.

media ID
An identifier that is written on a volume as part of the label.

Media Manager
Software that is part of NetBackup and manages the storage devices and removable media.

Media Manager Host
A host where Media Manager software is installed.

media server
NetBackup master or slave server to which the job (client) is sending the data.

menu interface
A character-based interface for use on terminals that do not have X Window capabilities.

mount
Make a volume available for reading or writing.

mount point
The point where a file system on a disk logically connects to a system's directory structure so the file system is available to users and applications.

MPX
See "multiplexing."

mtime
The point in time when a UNIX file is modified.

multiplexing
The process of sending concurrent-multiple backups from one or more clients to a single storage device and interleaving those images onto the media.

multiplexed group
A set of backups that were multiplexed together in a single multiplexing session.

NDMP
Network data management protocol.

NetBackup Client service
NetBackup Windows NT service that runs on clients and servers and listens for connections from NetBackup servers and clients in the network. When a connection is made, this service starts the necessary programs.

NetBackup configuration options
On UNIX servers and on UNIX and Macintosh, clients, these settings are made in the `bp.conf` file. On NetWare target and OS/2 clients, they are in the `bp.ini` file. On Windows NT servers and Microsoft Windows clients, these settings are called properties and are made through the Backup, Archive, and Restore interface or the Configure - NetBackup window in the administration interface.

NetBackup Database Manager service
NetBackup Windows NT service that runs on the master server and manages the NetBackup internal databases and catalogs. This service must be running on the master server during all NetBackup administrative operations.

NetBackup Device Manager service
The NetBackup Windows NT service that runs on a master or slave server and starts the robotic control processes and controls the reservation and assignment of volumes. This service exists only if the server has devices under Media Manager control. The process is `ltid`.

NetBackup properties
Same as NetBackup configuration options but are called NetBackup properties on Windows platforms.

NetBackup Request Manager service
The NetBackup Windows NT service that runs on the master server and starts the scheduler and receives requests from clients.

NetBackup Volume Manager service
The NetBackup Windows NT service that runs on a master or slave server and finds requested volumes. The process is `vmd`.

NIS
Network information service.

NFS
Network file system.

nonrobotic
See “standalone".

ODL
Optical disk library.

OSF and Motif
A set of specifications for user-interface design.

partitions
The logical partitions into which a magnetic disk is divided.

patch
A program that corrects a problem or adds a feature to an existing release of software.

path length
Number of characters in a pathname.

pathname
The list of directories in the path to a destination directory or file.

PC Clients
Refers to NetBackup clients that have Microsoft Windows (NT, 98, 95), Macintosh, or IBM OS/2 operating systems.

peername
The name by which a computer identifies itself when establishing connections to other systems.

port
A location used for transferring data in or out of a computer.

primary copy
The copy of an image that NetBackup uses to satisfy restores. When NetBackup duplicates an image, the original is designated as the primary copy.

privileges
The tasks or functions that a user, system, or application is authorized to perform.

progress report
Log where NetBackup records events that occur during user operations.

QIC
Quarter-inch-cartridge tape.

queued job
A job that has been added to the list of jobs to be performed.

raw-partition backup
Bit-by-bit backup of a partition of a disk drive on UNIX. On Windows NT, this is called a disk-image backup.

rbak
The program that DomainOS clients use to read data from tape during a restore.

registry
A Microsoft Windows NT, 98, and 95 database that has configuration information about hardware and user accounts.

restore
The process of restoring selected files and directories from a previous backup and returning them to their original directory locations (or to an alternate directory).

retention level
An index number that corresponds to a user-defined retention period. There are 10 levels from which to choose (0 though 9) and the retention period associated with each is configurable. Also see “retention period."

retention period
The length of time that NetBackup keeps backup and archive images. The retention period is specified on the schedule.

root
The highest level directory in a hierarchical directory structure. In MS-DOS, the root directory on a drive is designated by a backslash (for example, the root on drive C is C:\). On UNIX, the root directory is designated by a slash (/).

SCSI
Small computer system interface. This is a type of parallel interface that is frequently used for communicating with storage peripherals.

slave server
NetBackup servers whose attached storage units provide additional storage. The master server schedules and manages the backup and restore operations.

server directed restore
Using the client interface on the master server to restore files to any client. Only the administrator can perform this operation.

server independent restore
Restoring files by using a NetBackup server other than the one that was used to write the backup.

server list
The list of servers that a NetBackup client or server refers to when establishing or verifying connections to NetBackup servers. On a Windows NT server and Microsoft Windows clients, the list is in the NetBackup Configuration dialog box in the user interface on the client (see the NetBackup user’s guide for the client). On a UNIX server and UNIX and Macintosh clients, the list is in the `bp.conf` file. On NetWare target and OS/2 clients, the list is in the `bp.ini` file.

session
An instance of NetBackup checking its schedules for backups that are due, adding them to its worklist, and attempting to complete all jobs in the worklist. For user backups and archives, a session usually consists of a single backup or archive.

standalone
A qualifier used with drives and media to indicate they are not associated with a robot. For example, a standalone tape drive is one where you must manually find and insert tapes before using them. A standalone volume is one that is located in a standalone drive or is stored outside of a drive and designated as standalone in the volume configuration.

status code
A numerical code, usually accompanied by a message, that indicates the outcome of an operation.

storage migrator
Refers to the VERITAS Storage Migrator line of hierarchical storage management products for UNIX and Windows NT. These products make extra room on a disk by transparently moving data to other storage and then transparently retrieving the data when it is needed by a user or application.

storage unit
Refers to a storage device where NetBackup or Storage Migrator stores files. It can be a robot or consist of one or more single tape drives that connect to the same host.

SUSPENDED media state
If a volume is SUSPENDED, NetBackup can restore from it but cannot use it for backups. NetBackup retains a record of the Media ID until the last backup image on the volume expires.

symbolic link
On a UNIX system, this is a pointer to the name of the file that has the source data.

tape format
The format that an application uses to write data on a tape.

tape marks
A mark that is recorded between backup images on a tape.

tape overhead
The space required for data that is not part of the backup images. For example, tape marks and catalogs of what are on the tape are considered overhead.

tape spanning
Using more than one tape to store a single backup image.

tar
Tape ARchive program that NetBackup uses to extract backup images during a restore.

TC4 - Tape Carousel 4MM
A Media Manager designation for a category of robot. For the specific vendor types and models in this category, see the NetBackup release notes.

TC8 - Tape Carousel 8MM
A Media Manager designation for a category of robot. For the specific vendor types and models in this category, see the NetBackup release notes.

TLD - Tape Library DLT
A Media Manager designation for a category of robot. For the specific vendor types and models in this category, see the NetBackup release notes.

TLH - Tape Library Half-inch
A Media Manager designation for a category of robot. For the specific vendor types and models in this category, see the NetBackup release notes.

TLM - Tape Library Multimedia
A Media Manager designation for a category of robot. For the specific vendor types and models in this category, see the NetBackup release notes.

TL4 - Tape Library 4MM
A Media Manager designation for a category of robot. For the specific vendor types and models in this category, see the NetBackup release notes.

TL8 - Tape Library 8MM
A Media Manager designation for a category of robot. For the specific vendor types and models in this category, see the NetBackup release notes.

TIR
See “true image restore."

tpconfig
A Media Manager administration utility for configuring devices and is started from the command line. On UNIX, it has a character-based, menu interface that can be run from terminals that do not have X Windows capabilities.

transfer rate
The rate at which computer information is transferred between a source and a destination.

true image restore
Restores the contents of a directory to what it was at the time of any scheduled full or incremental backup. Previously deleted files are ignored.

TS8 - Tape Stacker 8MM
A Media Manager designation for a category of robot. For the specific vendor types and models in this category, see the NetBackup release notes.

TSH - Tape Stacker Half-inch
A Media Manager designation for a category of robot. For the specific vendor types and models in this category, see the NetBackup release notes.

user operation
A backup, archive, or restore that is started by a person on a client system.

verbose flag
Causes a higher level of detail to be written in the logs.

verify
An operation that compares the list of files that are actually on a volume with what NetBackup has recorded as being on it. The data that is on the media is not verified.

volume
Removable media that has been assigned a media ID and other attributes so Media Manager can use it.

volume pool
A set of volumes that are configured within Media Manager to be used by a single application and are protected from access by other applications and users.

wakeup interval
The time interval at which NetBackup checks for backups that are due.

wbak
The program that DomainOS clients use to write data on tape.

wildcard characters
A character that can be used to represent other characters in searches.

WORM media
Write once, read many (optical disks).

xbp
A backup, archive, and restore program for users on NetBackup UNIX clients. It has an X Windows based, graphical interface.

xbpadm
The main NetBackup administration utility on UNIX. It has an X Windows based graphical interface.

xbpmon
A NetBackup utility used for monitoring jobs on UNIX. It has an X Windows based graphical interface.

NetBackup 3.2 Media Manager Device Configuration Guide for UNIX

Index

A

ACS (see Automated Cartridge System)
Administrator, definition, xi
AIX (see RS6000)
Alternate client restores
    definition, 165
Alternate path restore
    definition, 166
Alternate target restore
    definition, 166
arch command, 24
Archive bit
    definition, 166
Archives
    definition, 166
AT&T style close, 37
autoconfig command, 123, 138
Automated Cartridge System
    on Sun4/SPARC, 27

B

Backup frequency
    definition, 170
Backup images, definition, 171
Backup windows
    definition, 166
Backups
    definition, 166
Berkeley-style close
    on HP9000-700, 86
    on HP9000-800, 101
    on Sun4/SPARC, 37
boot -r
    on Sun4/SPARC, 52
bp.conf file
    definition, 167

C

Cautions
    variable mode devices
        on SGI IRIX, 114
chdev command, 80
Client, NetBackup
    definition, 167
    user interface, 167
Command summary
    for DEC Alpha, 147
    for HP9000-700, 92
    for HP9000-800, 106
    for RS6000, 79
    for SGI IRIX, 123, 138
    for Sun4/SPARC, 51
    for SunOS, 24
config command, 25
Corsair, HP (see HP 1.2 GB optical)

D

DAT (see HP 4-mm DAT)
DEC Alpha
  introduction, 139
  locate-block, 143
  RS-232 robotic controls, 142
  SCSI robotic controls, 140
    example device files, 141
    make device files, 140
  tape drive configuration
    example device files, 144
    make device files, 143
DEC DLT (see Quantum DLT)
Device drivers
  ovpass, 56
  sd
    on Sun4/SPARC, 50
    on SunOS, 22
  SG
    on Sun4/SPARC, 33
    on SunOS, 2
  spt (SCSI pass through)
    on HP9000-800, 96
  st, 41
Disk-image backup
  definition, 169
DLT, definition, 169
dmesg command, 24

E

Exabyte compression drives
  on SGI IRIX, 116
  on SunOS, 13
Extended file marks, 65

F

Fast-tape positioning (see locate-block)
Fixed mode devices, 39, 65
Forward-space-record, 37

H

HP 1.2 GB optical
  on RS6000, 56, 73
  on SunOS, 21
HP 4-mm DAT
  on RS6000, 70
  on SGI IRIX, 117, 133
  on Sun4/SPARC, 45
  on SunOS, 15
HP C1560B DAT Autoloader
  logical unit number, 10
  on RS6000, 70
  on SGI IRIX, 117, 133
  on Sun4/SPARC, 45
  on SunOS, 15
HP optical disk
  on HP9000-700, 90
  on HP9000-800, 105
  on RS6000, 73
  on SGI IRIX, 122, 137
  on Sun4/SPARC, 48
  on SunOS, 20
HP9000-700
  introduction, 81
  optical disk configuration
    example device files, 90
    make device files, 90
  RS-232 robotic controls, 84
  SCSI robotic controls, 82
    example device files, 83
    make device files, 82
  SCSI, core I/O, 82
  SCSI, on board, 82
  tape drive configuration
    Berkeley-style close, 86
    example device files, 87
    make device files, 86
HP9000-800
  introduction, 93
  optical disk configuration
    example device files, 105
    make device files, 105

NetBackup 3.2 Media Manager Device Configuration Guide for UNIX

RS-232 robotic controls, 100
SCSI robotic controls, 94
example device files, 97, 99
make device files, 96, 98
spt (SCSI pass through) driver, 96
SCSI, core I/O, 98
SCSI, on board, 98
tape drive configuration
Berkeley-style close, 101
example device files, 102
make device files, 101
HSM, 171

I

IBM RS6000 (see RS6000)
install_ovpass script, 56, 79
install_path, 171
ioscan command
for HP9000-700, 92
on HP9000-800, 106
IRIX (see SGI IRIX)

K

Kernel changes
DEC Alpha, 145
SGI IRIX
for Exabyte 8900 compression, 116
for HP 4-mm DAT drives, 117, 133
for Quantum DLT, 119
SunOS
for Exabyte compression, 13
for HP 4-mm DAT drives, 15
for Quantum DLT, 18
for STK drives, 17

L

Library, definition, 172
locate-block
on DEC Alpha, 143
on RS6000, 66
on SGI IRIX, 114, 135
on Sun4/SPARC, 37
Logical unit numbers
on HP C1560B, 10
on Sun4/SPARC, 41
on SunOS, 9
lsdev command
on DEC Alpha, 147
on HP9000-800, 106
on RS6000, 79
lun (see logical unit numbers)

M

make command, 25
MAKEDEV
on DEC Alpha, 147
on SGI IRIX, 123, 138
on SunOS, 24
Media Manager host
definition, 173
mediad command, 110, 128
mkdev command, 79
mknod command, 147
mksf command, 92
modinfo command, 51
modstat command, 24
modunload command, 24
motd file, 24
multiple logical unit numbers, 42

N

NetBackup Client service
definition, 173
NetBackup configuration options, 173
NetBackup Database Manager service
definition, 174
NetBackup Device Manager service
definition, 174
NetBackup Request Manager service

NetBackup 3.2 Media Manager Device Configuration Guide for UNIX

definition, 174
NetBackup Volume Manager service
definition, 174
Nonrobotic, definition, 174

O

Optical disk
on HP9000-700, 90
on HP9000-800, 105
on RS6000, 73
on SGI IRIX, 122, 137
on Sun4/SPARC, 48
on SunOS, 20
ovpass driver, 56

P

Pass-through driver (see SCSI)

Q

Quantum DLT
on SGI IRIX, 119
on SunOS, 18

R

Raw partition backups
definition, 175
rem_drv command, 29, 51
remove_ovpass command, 79
Restores
definition, 175
Retention period
definition, 176
Robotic controls
RS-232
on DEC Alpha, 142
on HP9000-700, 84
on HP9000-800, 100
on RS6000, 62
on SGI IRIX, 113, 130
on Sun4/SPARC, 36
on SunOS, 6
SCSI
on DEC Alpha, 140
on HP9000-700, 82
on HP9000-800, 94
on RS6000, 57, 62
on Sequent, 154
on SGI IRIX, 111, 129
on Sun4/SPARC, 33
on SunOS, 2
root, 176
RS6000
introduction, 53
adapter numbers, 55
install_ovpass driver
loading, 56
unloading, 56
install_ovpass script, 56
locate-block, 66
optical disk configuration
make device files, 73
example device files, 74
set in nonvolatile memory, 75
ovpass driver, 56
pass-through driver, 56
RS-232 robotic controls, 62
SCSI pass-through driver, 56
SCSI robotic controls, 57, 62
example device files, 59
make device files, 57, 62
smit, 53
tape drive configuration
example device files, 68
extended file marks, 65
HP 4-mm DAT, 70
make device files, 66
multiple densities, 69
Sony AIT, 70
variable mode devices, 65

NetBackup 3.2 Media Manager Device Configuration Guide for UNIX

S

scip SCSI controller, 109, 127
Scripts
  /etc/rc.local (SunOS), 4
  install_ovpass, 56
  sg.install
    on Sun4/SPARC, 29
    on SunOS, 4
  sgscan, 5, 13, 25, 33, 34, 38, 44, 46, 48, 52
  spscan, 25
  vmconf, 25
SCSI
  integral, on SGI IRIX, 108, 126
  kernel ID mapping
    on SunOS, 8
  logical unit numbers
    on Sun4/SPARC, 41
    on SunOS, 9
  on board
    on HP9000-700, 82
    on HP9000-800, 98
    on SGI IRIX, 108, 126
  pass-through driver
    on RS6000, 56
    on Sun4/SPARC, 29
  robotic control
    on DEC Alpha, 140
    on HP9000-700, 82
    on HP9000-800, 94
    on RS6000, 57, 62
    on Sequent, 154
    on SGI IRIX, 111, 129
    on SunOS, 2
  spt (SCSI pass through) driver
    on HP9000-800, 96
scsi_command
  on RS6000, 77, 80
  on Sun4/SPARC, 49, 52
  on SunOS, 22
sd driver
  on Sun4/SPARC, 50
  on SunOS, 22
Server directed restore
  definition, 176
Session, NetBackup, 176
SG driver
  on Sun4/SPARC, 33
  on SunOS, 2
  rem_drv, 29
sg.build command, 51
sg.install script
  on Sun4/SPARC, 29, 51
  on SunOS, 4, 24
SGI IRIX
  introduction, 107, 125
  kernel changes
    for Exabyte 8900 compression, 116
    for HP 4-mm DAT, 117, 133
    for Quantum DLT, 119
  locate-block, 114, 135
  mediaid command, 110, 128
  optical disk configuration
    example device files, 122, 137
    make device files, 122, 137
  RS-232 robotic controls, 113, 130
  SCSI
    integral, 126
    on-board, 126
  SCSI robotic controls, 111, 129
    example device files, 112, 130
    make device files, 111
  tape drive configuration
    Exabyte compression, 116
    example device files, 115, 132
    HP 4-mm DAT, 117, 133
    make device files, 114, 132
    Quantum DLT, 119
    variable mode, 114
  using scip controllers, 109, 127
smit tool, 53
Sony AIT
  on RS6000, 70
  on Sun4/SPARC, 45

SPARC
running Solaris, 27
spt driver (see SCSI)
st driver, 41
st.conf.c file, 40
st_conf.c file, 13
ST_KNOWS_EOD attribute, 15
Standalone
definition, 177
stdef.h file, 16
STK drives, 17
Storage Migrator, 177
Storage units
definition, 177
Sun4/SPARC, 29
introduction, 27
adapter card removal, 27
locate-block, 37
optical disk configuration
example device files, 49
make device files, 48
set in nonvolatile memory, 49
RS-232 robotic controls, 36
SCSI pass-through driver, 29
SCSI robotic controls, 33
SG driver
loading, 29
unloading, 29
sg.install script, 29
st driver, 41
st.conf.c file, 40
tape drive configuration
Berkeley-style close, 37
example device files, 38, 46
HP 4-mm DAT, 45
make device files, 38
Sony AIT, 45
variable mode devices, 39
using ACS, 27
SunOS
kernel changes
for Exabyte compression, 13
for HP 4-mm DAT, 15
for Quantum DLT, 18
for STK drives, 17
kernel, rebuilding, 22
optical disk configuration
example device files, 21
make device files, 20
set in nonvolatile memory, 21
RS-232 robotic controls, 6
SCSI ID mapping
introduction, 8
change device assignment, 9
check device assignment, 9
finding conf file, 8
logical unit numbers, 9
SCSI robotic controls, 2
loading SG driver, 3
SG driver
loading, 4
unloading, 4
sg.install script, 4
st_conf.c file, 13
ST_KNOWS_EOD attribute, 15
stdef.h file, 16
tape drive configuration
make device files, 11
Exabyte compression, 13
example device files, 12
HP 4-mm DAT, 15
Quantum DLT, 18
STK drives, 17
Switch settings, HP 4-mm DAT
on DEC Alpha, 145
on HP9000-700, 89
on HP9000-800, 104
on RS6000, 70
on SGI IRIX, 117, 133
on Sun4/SPARC, 45
on SunOS, 15
Switch settings, Sony AIT
on RS6000, 70
on Sun4/SPARC, 46

NetBackup 3.2 Media Manager Device Configuration Guide for UNIX

T

Tape drive configuration
- on DEC Alpha, 143
- on HP9000-700, 86
- on HP9000-800, 101
- on RS6000
  - HP 4-mm DAT, 70
  - make device files, 66
  - Sony AIT, 70
- on SGI IRIX
  - Exabyte compression, 116
  - HP 4-mm DAT, 117, 133
  - make device files, 114, 132
  - Quantum DLT, 119
- on Sun4/SPARC
  - HP 4-mm DAT, 45
  - make device files, 38
  - Sony AIT, 45
- on SunOS
  - make device files, 11
  - Exabyte compression, 13
  - HP 4-mm DAT, 15
  - Quantum DLT, 18
  - STK, 17

tpconfig, definition, 178

True image restore
- definition, 178

V

Variable-mode devices
- on RS6000, 65
- on SGI IRIX, 114
- on Sun4/SPARC, 39

vmconf script, 25

vmunix file, 23

Volume pool
- definition, 179
- definition, 179

W

WORM media